Security Aspects of Distance-Bounding Protocols

Vom Fachbereich Informatik der
Technischen Universität Darmstadt
genehmigte
Dissertation
zur Erlangung des Grades
Doktor rerum naturalium (Dr. rer. nat.)
von

MSc. Maria Cristina Onete
geb. in Iași, Rumänien

Referenten: Prof. Dr. Marc Fischlin
Prof. Dr. Stefan Katzenbeisser

Tag der Einreichung: 20. Juni 2012
Tag der mündlichen Prüfung: 04. Juli 2012

Darmstadt, 2012
Hochschulkennziffer: D 17

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit —abgesehen von den in ihr ausdrücklich genannten Hilfen— selbstständig verfasst habe.

_____

Curriculum Vitæ

Cristina Onete

Personal

e-mail | cristina.onete@gmail.com

website | www.onete.net

Professional Experience

2009 – 2012 | PhD student at the CASED (TU Darmstadt), Germany, scholarship funded by LOEWE. Topic: "Security Aspects of Distance-bounding Protocols". Supervisor: Prof. Dr. Marc Fischlin.

2011 – 2012 | Founder and organiser of the first two editions of the CrossFyre workshop for female researchers in cryptography, aiming to promote young female students working in various areas of cryptography. This event was submitted as a proposal to the Marinna van Damme grants given by the TU Eindhoven, and it was chosen for the final selection round (3 participants were chosen out of over 40 proposals). In a follow-up of this selection round, the TU Eindhoven and CASED in Darmstadt co-funded the organisation of the first two editions of CrossFyre, the first organised (by me) in Darmstadt in 2011, and the second organised in Eindhoven in 2012 (an event where I was one of two co-organisers, assisting the main organiser). This event will be organised again in 2013, and I am again part of the organisation.

2010 – 2012 | Reviewer for cryptographic conferences, including: Asiacrypt (2010, 2012), SCN 2010, CT-RSA (2010, 2011), ProvSec 2011, Eurocrypt (2011, 2012).

2006 – 2007 | Student Research and Teaching Assistant, Eindhoven University of Technology.

Academic Publications

RFID-TA 2012 (Accepted) | Mafia Fraud Attack against the RC Distance-Bounding Protocol
with: *Katerina Mitrokotsa, Serge Vaudenay*; IEEE RFID-TA 2012, Proceedings will follow.

MPICC 2012 (Accepted) | RFID Distance-Bounding: What is Wrong and How to Fix it
with: *Marc Fischlin*; 5th MPICC Interdisciplinary Conference on Current Issues in IT Security, 2012, Proceedings will follow.

ePrint, 2012 | Key Updates for RFID Distance-Bounding Protocols: Achieving Narrow-Destructive Privacy, Submitted to SCN 2012,
ePrint, report 165/2012

ePrint, 2012 | Provably Secure Distance-Bounding: an Analysis of Prominent Protocols
with: *Marc Fischlin*, Submitted to RFIDSec 2012,
ePrint, report 128/2012

ISC 2011 | A Formal Approach to Distance-Bounding RFID Protocols
with: *Ulrich Dürholz, Marc Fischlin, Michael Kasper*
LNCS 7001, pp. 47-64

ACNS 2011	Relaxed Security Notions for Signatures of Knowledge with: *Marc Fischlin* LNCS 6715, pp. 309-326
ePrint, 2011	Security & Indistinguishability in the Presence of Traffic Analysis with: *Daniele Venturi* ePrint, report 260/2011
ACNS 2010	Redactable Signatures for Tree-Structured Data: Definitions and Constructions with: *Christina Brzuska, Heike Busch, Oezguer Dagdelen, Marc Fischlin, Martin Franz, Mark Manulis, Andreas Peter, Bertram Poettering, Dominique Schröder* LNCS 6123, pp. 87–104
ECCTD 2011	Finding spanning trees and Hamiltonian circuits in an un-oriented graph: an algebraic approach with: *Cristian Onete*, IEEE Conference Publications, doi: 10.1109/ECCTD.2011.6043384, pp. 453-456
EUROCON 2011	A novel condition for Hamiltonicity; constructing Hamiltonian Circuits with: *Cristian Onete*, IEEE Conference Publications, doi: 10.1109/EUROCON.2011.5929160, pp. 1-4
SM2ACD 2010	Enumerating all the spanning trees in an un-oriented graph – a Novel approach with: *Cristian Onete*, IEEE Conference Publications, doi: 10.1109/SM2ACD.2010.5672365, pp. 1-5
ISCAS 2010	Indefinite Matrices of Linear Electric Circuits, Their Pseudoinverses, and Applications in Related Fields with: *Cristian Onete*, IEEE Conference Publications, doi: 10.1109/ISCAS.2010.5537168, pp. 2402-2405

University Education

2006 – 2008	Studies of Industrial Mathematics at the Eindhoven University of Technology, Master finished Cum Laude in 2008 with degree "MSc.", title of the diploma thesis, "Visualisation of Modern Key Exchange Schemes for More than Two Parties in CrypTool and their Security Analysis".
2003 – 2007	Studies of Applied Mathematics at the Eindhoven University of Technology, Bachelor finished in 2007 with degree "Ingenieur", title of the thesis: "Linear Representations of Finite Groups; Some Properties of Braids."

Projects

2007 – 2008	Internship project: "Elliptic Curves and Pairing Based Cryptosystems". Implemented pairing based protocol and equivalent finite field protocol. Efficiency analysis and improvements to Desmedt Lange protocol.
2006 – 2007	Implemented part of a statistical software which models reliability. Student research assistant under supervision of Prof. A. di Bucchianico.

Teaching

09.2008 – 10.2008	Exercise course "Calculus I" for Technology Management students at TU/e, given in Dutch.
09.2007 – 01.2008	Exercise courses "Calculus I", "Calculus II" for Technology Management students at TU/e, given in Dutch.
09.2006 – 02.2007	Exercise courses "Calculus I", "Calculus II", and "Calculus III" for Technology Management students at TU/e, given in Dutch.
03.2006 – 05.2006	Exercise course "Linear Vector Spaces" for Mathematics students at TU/e, given in Dutch.
2003 – present	Tutoring and e-tutoring high-school students from the ISSE in Eindhoven. Subjects: Mathematics, Chemistry.

Language Skills

Fluent in English, Dutch, Spanish, German, and Romanian.
Also speak limited Italian.

Computer Skills

Knowledge of C++, Java, Mathematica, R, LaTex

Personal Interests

Walking/hiking, writing (fantasy novels), football, Role-playing, cooking, languages, people.

Darmstadt, 21st of August, 2012.

Abstract

Authentication protocols, run between a so-called prover and a so-called verifier, enable the verifier to decide whether a prover is legitimate or not. Such protocols enable access control, and are used in e.g. logistics, public transport, or personal identification. An authentication protocol is considered secure if an adversary cannot impersonate a legitimate prover. Such an adversary may eavesdrop authentication attempts between a legitimate prover and a legitimate verifier, interact with either of the two honest parties, or perform a man-in-the-middle (MITM) attack, but without purely relaying messages between the honest parties (see Figure 2 (a)).
Distance-bounding is a feature that enables authentication protocols to also withstand MITM *relay attacks*, where an adversary forwards information between the prover and verifier such that neither honest party is aware of the attack. The goal of the adversary is to be authenticated by the verifier as a legitimate prover. In practice, the adversary consists of two parties, a *leech*, which impersonates the verifier to the prover, and a *ghost*, which impersonates the prover to the verifier. This is also depicted in Figure 2. Following the initial paper by Desmedt [24], pure relay attacks are called *mafia fraud*.

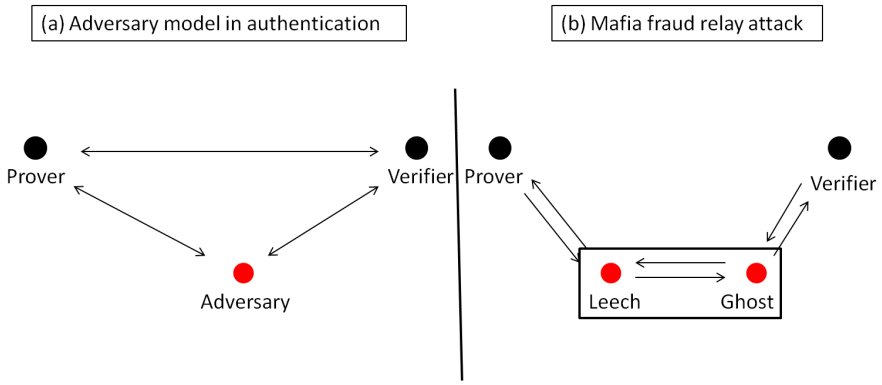

Figure 1: Adversary models in (a) authentication and (b) mafia fraud.

In distance-bounding protocols, introduced by Brands and Chaum in 1993 [13], a clock is mounted on the verifier, such that it can measure the time-of-flight between sending a challenge and receiving a response. Following the idea that pure relay introduces a processing delay in the MITM adversary, a verifier now compares the measured time-of-flight with a pre-set value $t_{\max}$ (in practice, an upper bound associated with the maximum trusted communication distance). If the communication speed is constant (and very fast), the verifier authenticates the prover if (i) the verifier is convinced that the prover is legitimate, and (ii) the prover is within the maximum distance associated with $t_{\max}$. Such protocols were recently implemented by Rasmussen and Čapkun [68] and Ranganathan et al. [66].
In most distance-bounding protocols in the literature, the prover-verifier communication is run in multiple rounds, or *phases*, which are classified as either *time-critical* if the verifier's clock measures the roundtrip time, or *lazy* if the clock is not used. Note, however, that e.g. the protocol due to Rasmussen and Čapkun [69] is not round-based.
There are three classical attacks that distance-bounding protocols should in general address:

1. **Mafia Fraud:** Here, an adversary attempts to authenticate to the verifier in the presence of an honest prover (however, the verifier's clock prevents pure relay). Both honest parties are unaware of the attack.
2. **Terrorist Fraud:** Here the dishonest prover provides some limited help to the adversary, such that the adversary is able to authenticate to the honest verifier. However, the prover should not forward any information that allows the adversary to authenticate without the prover's help. Intuitively, a terrorist fraud attack is successful if the adversary is successful in authenticating with the prover's help, but not without it.
3. **Distance Fraud:** The adversary in this scenario is a dishonest prover placed far from the verifier (i.e. outside the range associated with $t_{\max}$). The goal of the adversary is to authenticate, thus fooling the verifier's clock.

A fourth security notion (suggested much later by Avoine and Tchamkerten [6]) is that of classical impersonation security, where the adversary interacts with either the prover or the verifier (but not in a MITM attack) and wins if it authenticates successfully. This attack is particularly dangerous for implementations on resource-constrained devices, e.g. RFID tags, where the provers (tags) only support a small number of time-critical rounds. Thus, the mafia fraud resistance of the protocol (which hinges on the number of time-critical rounds) is very low.

The main contribution of this thesis is to introduce a unitary, three-layered, formal framework for proving security of distance-bounding protocols. The main layer of this model is the central layer and captures a static model, where the prover and verifier share a secret key, which is never updated. In this context, we define the fundamental threats to distance-bounding protocols, namely: mafia, terrorist, and distance fraud, and impersonation security. Additionally, we take a closer look at mafia and terrorist fraud resistance, extending mafia fraud to capture prover aborts, and describing three flavours of terrorist fraud resistance, depending on how much information a dishonest prover is willing to pass to the adversary. The two remaining layers of our model discuss privacy aspects of distance-bounding protocols: in our top layer we include location privacy, whereas our bottom layer concerns privacy in authentication.
A second very important contribution is to describe two constructions: the first provable-secure terrorist fraud resistant, and the first provably location-private distance-bounding protocol in the literature (see more details below). Also very important are the three compilers we describe, which enable black-box transformations from either one tier of the main framework to another, or from one degree of mafia fraud resistance to another. More concretely, we proceed in the following directions.

Fundamental Model. The central, fundamental layer of this framework includes the four standard attacks enumerated above, extending and formalizing previous work due to Avoine et al. [4]. The models account for exact bounds for the levels of mafia, terrorist, and distance fraud resistance, and for impersonation security of any round-based distance-bounding protocol; having compared the notions, we prove them to be independent, contrary to popular belief and to the results of [4]. As this is the fundamental layer of our model, we consider it paramount to assess the security properties of several distance-bounding protocols in the literature. We choose the more prominent schemes of Brands and Chaum [13], Hancke and Kuhn [42], Avoine and Tchamkerten [6], Kim and Avoine [49], Reid et al. [70], Bussard and Bagga [15], and the Swiss-Knife protocol due to Kim et al. [50]. In so doing we give a generic attack against the terrorist fraud resistance of the schemes in [15, 50, 70], showing that they attain a different, in many ways more relaxed notion of terrorist fraud resistance.
The central layer of our model considers static keys only. However, we extend this scenario to also include privacy aspects, both in authentication (the bottom layer of our model) and with respect to the prover's location (the upper layer of the model). Here, we define privacy in authentication in the sense of Vaudenay [73], i.e. distance-bounding sessions should be unlinkable. The adversary in this scenario is also allowed to corrupt provers and learn their internal states (i.e. the secret key); depending on the adversary strength, the adversary may then be able to interact with other provers or not. If the adversary is unable to learn the result of an authentication protocol between a prover and the verifier, we speak of *narrow* privacy; else, we speak of *wide* privacy. Though Vaudenay's model is well studied for RFID authentication, it has not been applied to distance-bounding.
In order to achieve privacy, we extend the fundamental, static model in our central layer to account for key updates, such that the prover and verifier use different keys to generate their responses during each protocol run. We also give compilers which preserve the static-key model properties of a scheme (i.e. their levels of mafia and distance fraud resistance, and impersonation security) in the key-update model, furthermore transforming narrow-weak private distance-bounding protocols (in the sense of [73]) into narrow-destructive private schemes. Recall that narrow privacy refers to the fact that the adversary does not learn the verifier output for protocol runs. Note that in the symmetric-key scenario this is the most one can achieve, since [73] proves that narrow-strong privacy requires key agreement. In our compiler, the prover updates state early, after an initial lazy authentication phase, whereas the verifier updates state only after authenticating the prover. Furthermore, the verifier can "catch up" with the prover by means of an initial state-recognition phase. However, in order to ensure that the verifier does not enter an infinite iterative process, resulting in a denial of service (DoS) attack, we split the prover state into two parts, one which is updated at each authentication attempt, and another which is only updated upon verifier authentication. This latter state fragment then enables the verifier to bypass DoS attacks.
The upper layer of our model, location privacy, was already addressed in the literature by Rasmussen and Čapkun [69]. However, they did not formally define their security model and in fact, we can prove that their proposed scheme is *not* location private in our definition. Note that, as opposed to authentication, where location privacy is generally used to mean that no location data is leaked by provers in authentication sessions, in distance bounding provers necessarily leak one piece of information, namely whether they are in the verifier's proximity or not. For location privacy we require that no further information can be leaked from provers which are within proximity of the verifier. We show an impossibility result: namely that location privacy cannot be achieved in an information-theoretical sense, and furthermore it cannot even be computationally attained for strong adversaries, which are in practice able to triangulate signals and gauge signal strength. We also show how to achieve location privacy in a weaker model, at the cost of a larger communication complexity; in particular, our construction is a modification of the construction in [69].
We stress that we view distance bounding as an extension of authentication. In particular, we consider mafia and distance

fraud resistance (as well as impersonation security) to be basic requirements of distance-bounding protocols. This contradicts the idea used by Rasmussen and Čapkun in their construction [69]: indeed, their protocol only attains distance fraud resistance. Our location private protocol in fact makes a modification in the original scheme in [69], so as to thwart a mafia fraud attack against this scheme. We also describe this attack.

Mafia and Terrorist Fraud. Having completed the three-tiered description of our model, we also look more in depth at mafia and terrorist fraud resistance. We show that, unlike previous frameworks, our model also captures so-called key-learning attacks, where the adversary interferes in a few time-critical rounds of otherwise honest prover-verifier communication, with the goal of learning some bits of the secret key. Such attacks are known for authentication, but they have never been considered in the distance-bounding setting. We also model an extension of key-learning attacks, which could appear in practice in the case of prover aborts. For the latter notion of terrorist fraud resistance we elaborate on our previous definition of terrorist fraud resistance. We explain that though most protocols in the literature do not attain this strong, simulation-based notion, our notion *can* be achieved (and we describe a protocol that achieves it). We also define a game-based flavour of terrorist fraud resistance, which is the one most commonly achieved by protocols in the literature. The difference between the notions is roughly this. In our simulation-based model the prover may leak some information about his secret key to the adversary, as long as this information does not enable a simulator to authenticate *with equal probability* in the future. In practice, the prover can then choose to leak only very little information about the secret key, thus ensuring that (i) the adversary does authenticate while the prover helps, and (ii) once the prover stops helping, the adversary's probability to succeed drops. By contrast, in the game-based mode, the prover is unwilling to leak *any* information to the adversary: thus, the adversary is far more limited.
More in detail, for the topic of mafia fraud resistance, our model extends previous mafia fraud models to account for key-learning attacks as outlined above. Such attacks are particularly useful against protocols aiming to achieve terrorist fraud resistance. We also introduce an extension to the mafia fraud definition (called strong mafia fraud strMF) to account for adversaries that can hijack honest authentication sessions and thus authenticate from an aborting prover. We also show that strMF security is strictly stronger than mafia fraud resistance.
We also note that the simulation-based model of terrorist fraud resistance SimTF is, on the one hand, very strong (in fact we prove that some prominent schemes in the literature addressing this attack are insecure in our framework). On the other hand, however, this model does not cover all adversarial capabilities (since the prover only interacts with the adversary offline). Thus, we introduce both a stronger simulation-based notion strSimTF and a more relaxed, game-based notion GameTF security. The former extension provides an even higher degree of security, whereas the latter allows for more efficient constructions (while at the same time eliminating obvious attacks). We prove the well-known construction of Bussard and Bagga [15], which is SimTF *in*secure, to be GameTF secure. In fact, most protocols in the literature claiming to achieve terrorist fraud resistance, are actually GameTF secure, and not SimTF secure. We argue that whereas the GameTF notion seems to achieve a minimal degree of terrorist fraud resistance, it is very restrictive and may enable attacks, particularly in scenarios where the dishonest prover is prepared to yield a few bits of the secret key if the result is that the adversary can then authenticate.

Constructions and Compilers. As far as constructions are concerned, we (i) show the first provably secure SimTF-secure protocol, and show that it is also strSimTF-secure, and (ii) we design the first provably-secure location private protocol in the literature. Furthermore, we show compilers to turn mafia fraud resistant schemes into strMF secure constructions. Moreover, our compiler also works if the underlying protocol need not be fully maifa fraud resistant, but could be susceptible to key-learning attacks (which is a particular form of mafia fraud). This compiler uses a final lazy authentication step, which prevents adversaries from hijacking aborted prover-verifier sessions. Both compilers can be run, with several optimizations, on most distance-bounding constructions in the literature. We furthermore show several other handy tricks, both in the modelling department (where we show how to prove various notions in our model) and for construction purposes (we show how to easily achieve impersonation security and how mutual authentication can be used towards delegating some computational burden on the verifier).
Future topics include (i) modelling extended scenarios for distance-bounding protocols, with multiple provers and a single verifier, or multiple provers and multiple verifiers, formalizing previous work by Čapkun et al. [74]; (ii) investigating how far our results apply in a phase-less model (as the one considered for location privacy) or in a public-key setting; (iii) investigating a few relationships between the relations we defined and which have not yet been investigated, such as whether the key update compiler preserves GameTF security. Furthermore, we stress that tight security reductions are highly desirable, thus it is interesting to design compilers and constructions which achieve properties such as privacy, KLMF security, and strMF security with tight reductions. An important direction for future research is constructing distance-bounding protocols for public-key scenarios (our results are in they symmetric key setting). In particular, it is desirable to investigate the

applicability of distance-bounding to RFID tags which can perform elliptic curve (EC) computations. A final direction for future work is to achieve stable implementations of distance-bounding protocols for various architecture and investigating how far they can be used to prevent the attacks described in this framework.

Zusammenfassung

Protokolle zur Authentisierung sind Protokolle mit 2 Teilnehmern, einem sogenannten "Prover", dem Beweiser, der sich authentisieren will, und dem Verifizierer, der die Authentizität des Provers feststellen möchte. Protokolle dieser Art sind ein elektronisches Mittel der Zugangsbeschränkung und finden zum Beispiel Anwendung in Logistik, öffentlichem Nahverkehr oder persönlicher Identifikation. Ein Authentisierungsprotokoll nennen wir sicher, wenn ein Angreifer sich nicht als legitimer Prover ausgeben kann. Ein solcher Angreifer darf zu diesem Zweck Authentisierungsprotokolle des Provers beobachten und mit einem oder beiden interagieren, einen Man-In-The-Middle (MITM)-Angriff ausführen, allerdings ohne dass er ausschließlich Nachrichten zwischen dem Prover und dem Verifizierer weiterleitet (siehe auch Abbildung 2 (a)).
Distance-bounding (Abstandsbeschränkung, -approximierung) ist eine Eigenschaft von Authentisierungsprotokollen, die darüber hinaus MITM-Angriffe verhindert, bei denen nur Nachrichten zwischen dem Prover und dem Verifizierer weitergeleitet werden, ohne dass der Prover oder der Verifizierer den Angriff bemerken. Das Ziel des Angreifers ist es, sich gegenüber dem Verifizierer als legitimer Prover auszugeben. Um diesen Angriff tatsächlich auszuführen, teilt sich der Angreifer in zwei Teilnehmer aus, einen sogenannten "Leech", der dem Prover gegenüber die Rolle des Verifizierers einnimmt und einem "Ghost", der sich dem Verifizierer gegenüber als Prover ausgibt, siehe Abbildung 2. Gemäß dem ursprünglichen Artikel von Desmedt [24], heißen Angriffe, bei denen nur Nachrichten weitergeleitet werden, *Mafia Fraud*.

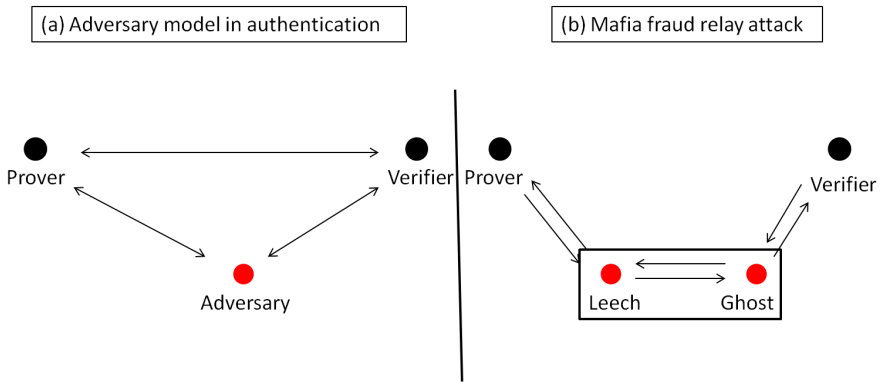

Figure 2: Adversary models in (a) authentication and (b) mafia fraud.

Distance-Bounding-Protokolle wurden 1993 von Brands und Cham [13] eingeführt, indem der Verifizierer eine Uhr implementiert, sodass er den Zeitabstand zwischen dem Senden einer Challenge und dem Empfangen einer Antwort messen kann. Dadurch, dass reines Weiterleiten durch den MITM-Angreifer eine Verzögerung verursacht, kann der Verifier nun den tatsächlichen mit dem erwarteten Zeitabstand $t_{\max}$ vergleichen (Diese obere Schranke $t_{\max}$ für den Zeitabstand entspricht dem größtmöglichen, erlaubten Abstand zwischen dem Prover und dem Verifizierer). Wenn Datenübertragung constante Zeit benötigt (und sehr schnell ist), dann authentisiert der Verifizierer den Prover, sofern (i) der Verifier sich von der Legitimität des Provers überzeugt hat, und (ii) der Prover sich innerhalb des mit $t_{\max}$ assoziierten Höchstabstandes befindet. Derartige Distance-Bounding-Protokollen wurden kürzlich von Rasmussen und Čapkun [68] und Ranganathan et al. [66] implementiert.
Die meisten existierenden Distance-Bounding-Protokolle führen die Prover-Verifizierer-Kommunikation in mehreren Runden aus, oder Phase, die entweder als *zeitkritisch* bezeichnet werden, wenn der Verifizierer die Gesamtzeit der Protokollausführung misst, oder *lazy*, wenn der Verifizierer seine Uhr nicht verwendet. Es gibt aber auch Protokolle, die nicht rundenbasiert sind. Hier ist zum Beispiel das Protokoll von Rasmussen and Čapkun [69] zu nennen.
Es gibt drei klassische Angriffe, die Distance-Bounding-Protokolle im Allgemeinen abdecken sollen:

1. **Mafia Fraud:** Hier versucht der Angreifer, sich dem Verifizierer gegenüber in der Anwesenheit des Provers zu legitimieren (Allerdings verhindert die Uhr des Verifizierers reine Weiterleitungsangriffe). Beide Teilnehmer, der Prover und der Verifizierer, sind sich des Angriffes nicht bewusst.
2. **Terrorist Fraud:** Hier unterstützt der unehrliche Prover den Angreifer auf beschränkte Weise, sodass der Angreifer sich dem Verifizierer gegenüber authentisieren kann. Die Hilfe des Provers ist allerdings insofern beschränkt, als dass er dem Angreifer keine Informationen zur Verfügung stellt, die es ihm ermöglicht, sich anschließend auch ohne weitere Hilfe des Provers dem Verifizierer gegenüber zu authentifizieren. Intuitiv ist ein Terrorist Fraud Angriff dann erfolgreich, wenn der Angreifer sich mithilfe des Provers authentisieren kann, aber nicht ohne dessen Hilfe.

3. **Distance Fraud:** In diesem Angriffsszenario ist der Angreifer ein unehrlich Prover, der vom Verifizierer weit entfernt befindet (das heißt, außerhalb des mit $t_{\max}$assoziierten Abstandes). Das Ziel des Angreifers ist es, sich zu authetisieren und die Uhr des Verifizierers zu täuschen.

Ein vierter Sicherheitsbegriff, der sehr viel später von Avoine und Tchamkerten [6] eingeführt wurde, ist der Angriff klassischer Impersonifizierungssicherheit, wo der Angreifer entweder mit dem Prover oder mit dem Verifizierer interagiert (aber nicht in einem MITM Angriff) und gewinnt, wenn er sich erfolgreich authentifiziert. Dieser Angriff ist besonders gefährlich für Implementierungen auf Ressource-beschränkten Geräten, z.B. RFID Tags, bei denen der Prover (Tag) nur eine kleine Anzahl zeitkritischer Runden unterstützt. Daher ist die Resistenz des Protokolls gegen Mafia Fraud Angriffe (die von der Anzahl der zeitkritischen Runden abhängt) sehr niedrig.
Der wichtigste Beitrag dieser Arbeit ist die Einführung eines unitären, dreischichtigen, formalen Frameworks, um die Sicherheit von Distance-Bounding-Protokollen zu beweisen. Die wichtigste Schicht dieses Frameworks ist die mittige Schicht, die ein statisches Modell beschreibt, in welchem der Prover und der Verifizierer einen gemeinsamen, geheimen Schlüssel teilen, der nicht aktualisiert wird. Vor diesem Hintergrund definieren wir die fundamentalen Angriffe gegen Distance-Bounding-Protokolle, nämlich: Mafia, Terrorist und Distance Fraud, und Impersonifizierungssicherheit. Darüber hinaus verfeinern wir die Definition der Resistenz gegen Mafia und Terrorist Fraud, in dem wir Mafia Fraud auf Angriffe ausweiten, bei dem der Angreifer eine Sitzung übernimmt wo der Prover abgebrochen hat. Die zwei verbleibenden Schichten unseres Modells beschäftigen sich mit Aspekten der Privatsphäre von Distance-Bounding-Protokollen: In unserer obersten Schicht betrachten wir die Geheimhaltung des Aufenthaltsortes, während unsere unterste Schicht sich mit der Geheimhaltung der Authentisierung selbst beschäftigt.
Ein weiterer, ebenfalls sehr wichtiger Beitrag ist die Beschreibung zweier Konstruktionen: Wir geben die erste Konstruktion für ein Protokoll, welches beweisbar sicher gegen Terrorist Fraud ist, und die erste Konstruktion, die beweisbar Geheimhaltung aller Informationen über Aufenthaltsort erhält (mehr Details weiter unten). Auch sehr wichtig sind die drei Compiler, die wir beschreiben, welche Black-Box-Transformationen von jeder der drei Schichten des Frameworks zu jeder anderen erreichen. Konkret sind unsere Beiträge wie folgt:

Fundamentales Modell Die mittlere, fundamentale Schicht dieses Frameworks deckt alle vier oben genannten Standardangriffe ab, was existierende Arbeit von Avoine et al [4] erweitert und formalisiert. Die Modelle beinhalten exakte Schranken für den Grad der Resistenz gegen Mafia, Terrorist, und Distance Fraud sowie für Impersonifizierungssicherheit für beliebige rundenbasierte Protokolle; wir vergleichen die Sicherheitsbegriffe und zeigen, dass keine die andere impliziert. Dies steht im Gegensatz zu einer weit verbreiteten berzeugung sowie der Arbeit [4]. Da dies die fundamentale Schicht unseres Modells ist, ist es von zentraler Bedeutung, die Sicherheit von existierenden Distance-Bounding-Protokollen in unserem Modell zu analysieren. Wir betrachten die promintenten Protokolle von Brands und Chaum [13], Hancke und Kuhn [42], Avoine und Tchamkerten [6], Kim und Avoine [49], Reid et al. [70], Bussard und Bagga [15], sowie das Schweizer-Messer-Protokoll von Kim et al. [50]. Hierbei zeigen wir einen generischen Angriff gegen die Resistenz gegen Terrorist Fraud der Protokolle in [15, 50, 70], wodurch wir zeigen, dass sie eine andere, in vielerlei Hinsicht schwächere Variante von Resistenz gegen Terrorist Fraud erreichen.
Die mittlere Schicht unseres Models betrachtet nur statische Schlüssel. Dahingegen erweitern wir das Angriffsszenario, um auch Aspekte der Privatsphäre miteinzuschließen, sowohl bezüglich Authentisierung (in der unteren Schicht unseres Modells) als auch bezüglich des Aufenthaltsortes des Provers (obere Schicht unseres Modells).
Hier definieren wir Geheimhaltung der Authentisierung im Sinne von Vaudenay [73], das heißt, verschiedene Distance-Bounding-Sitzungen sollen nicht in Verbindung gebracht werden können, also "unlinkable" sein. Der Angreifer in diesem Angriffsszenario darf Prover korrumpieren und ihren internen Zustand erfahren (das heißt, den secret key); anhängig von der Stärke des Angreifers kann der Angreifer zusätzlich mit anderen Provern interagiern, oder auch nicht. Wenn der Angreifer das Ergebnis eines Authentisierungsprotokolls zwischen einem Prover und einem Verifizierer nicht bestimmen kann, sprechen wir von *narrow privacy*; ansonsten bezeichnen wir die Eigenschaft als *wide privacy*. Vaudenays Modell ist gut erforscht bezüglich RFID Authentisierung, wohingegen es bislang nicht auf Distance-Bounding angewandt wurde.
Um Privacy zu erreichen, erweitern wir das fundamentale, statische Modell der mittleren Schicht, um auch Schlüsselaktualisierungen abzudecken, sodass der Prover und der Verifizierer verschiedene Schlüssel pro Protokollausführung verwenden können. Darüber hinaus konstruieren wir Compiler, die die Sicherheitseigenschaften des statischen Modells (also den Grad ihrer Resistenz gegen Mafia und Distance Fraud, und Impersonifizierungssicherheit) auch bei Schlüsselaktualisierungen erhalten. Darüber hinaus transformieren sie Distance-Bounding-Protokolle, die narrow-weak private sind (im Sinne von [73]) in Protokolle, welche narrow-destructive private sind. Wir erinnern uns daran, dass narrow privacy die Eigenschaft beschreibt, dass der Angreifer das Ergebnis von Protokollausführungen nicht erfährt. Für Verfahren mit symmetrischen Schlüsseln ist dies die stärkste Eigenschaft, die erreicht werden kann, da [73] zeigt, dass narrow-strong privacy einen sicheren Schlüsselaustausch bedingt. In unserem compiler aktualisiert der Prover seinen Zustand frühzeitig, nach der unitialen lazy

Authentisierungsphase, wohingegen der Verifizierer seinen Zustand er nach der Authentisierung des Provers aktualisiert. Darüber hinaus kann der Verifizierer den Prover "einholen", indem man zu Beginn eine Phase der Zustandserkennung ausführt. Um allerdings sicherzustellen, dass der Verifizierer nicht keine unendliche iterativen Schleife beginnt, sodass man einen "Denial of Service" (DoS) Angriff durchführen kann, teilen wir den Zustand des Provers in zwei Teile: Einen, der bei jedem Authentisierungsversucht aktualisiert wird und einen anderen, der nur aktualisiert wird wenn die Authentisierung erfolgreich ist. Das letztere Fragment des Zustandes ermöglicht dem Verifizierer, einen DoS-Angriff abzuwehren.
Die obere Schicht unseres Modells, Geheimhaltung des Aufenthaltsortes, wurde bereits von Rasmussen und Čapkun [69] betrachtet. Allerdings formulierten sie ihr Sicherheitsmodell nicht formal, sodass wir sogar zeigen können, dass das von ihnen vorgeschlagene Protokol *nicht* Geheimhaltung des Aufenthaltsortes erreicht. Man sollte beachten, dass bei Distance-Bounding im Gegensatz zu Authentisierung, wo Geheimhaltung des Aufenthaltsortes im Allgemeinen verwendet wird, um zu sagen, dass keine Informationen über den Aufenthaltsort preisgegeben werden, es so ist, dass Distance-Bounding notweniger-weise zumindest etwas Information über den Aufenthaltsort preisgibt, nämlich, ob der Prover sich in der Nähe befindet oder nicht. Für die Geheimhaltung des Aufenthaltsortes im Zusammenhang mit Distance-Bounding ist die Bedingung, dass keine darüber hinausgehende Information preisgegeben wird. Wir zeigen ein Unmöglichkeitsresultat: Geheimhaltung des Aufenthaltsortes kann nämlich nicht im informationstheoretischen Sinne erreicht werden, und darüber hinaus kann es auch im computational Sinne nicht erreicht werden, sofern der Angreifer stark ist, die Signale triangulieren und die Signalstärke bestimmen können. Wir zeigen ebenfalls, wie Geheimhaltung des Aufenthaltsortes in einem schwächeren Modell erreicht werden kann, wenn man bereit ist, die Kommunikationskomplexität zu erhöhen; insbesondere ist unserer Konstruktion eine Modifikation der Konstruktion aus[69].
Wir wollen hervorheben, dass Distance-Bounding für uns eine Erweiterung von Authentisierung ist. Insbesondere betrachten wir Resistenz gegen Mafia und Distance Fraud (als auch Impersonifizierungssicherheit) als grundlegende Anforderungen an Distance-Bounding-Protokolle. Dies widerspricht der Idee von Rasmussen und Čapkun und ihrer Konstruktion [69]: Tatsächlich erreicht ihr Protokoll nur Resistenz gegen Distance-Fraud. Unser Protokoll für die Geheimhaltung des Aufenthaltsortes modifiziert also das ursprüngliche Protokoll aus [69], um einen Mafia Fraud Angriff auf dieses Protokoll auszuschließen. Wir beschreiben diesen Angriff.

Mafia und Terrorist Fraud. Nachdem wir unser dreischichtiges Modell vollständig definiert haben, analysieren wir Mafia und Terrorist Fraud tiefergehend. Wir beweisen, dass unser Modell im Gegensatz zu vorherigen Modellen auch sogenannte key-learning (Schlüssel-lernende) Angriffe abdeckt, bei denen der Angreifer aktive Angriffe auf die wenigen zeitkritischen Runden einer sonst ehrlichen Prover-Verifizierer-Sitzung ausfhren darf. Das Ziel des Angreifers ist, einige wenige Bits des Schlüssels zu lernen. Diese Art des Angriffs sind im Zusammenhang mit Authentisierung schon bekannt. Im Zusammenhang mit Distance-Bounding sind sie jedoch bislang nicht betrachtet worden. Wir modellieren eine erweiterte Variante von Key-Learning Angriffen, welche praktisch relevant sind im Falle, dass der Prover die Ausfhrung des Protokolls vorzeitig beendet.
In Bezug auf Resistenz gegen Terrorist Fraud erweitern wir unsere ursprüngliche Definition von Resistenz gegen Terrorist Fraud. Wir zeigen, dass, obwohl die meisten Protokolle in der Literatur nicht resistet gegen Terrorist Fraud sind (bezüglich dieser simulationsbasierten Definition), es sehr wohl möglich ist, Resistenz gegen Terrorist Fraud gemäß dieser Definition zu erreichen. Insbesondere geben wir ein Protokoll an, welches Resistenz gegen Terrorist Fraud erreicht. Darüber hinaus geben wir eine spielbasierte Definition für Resistenz gegen Terrorist Fraud an, welche von den meisten existierenden Protokollen erreicht wird. Im Großen und Ganzen kann man den Unterschied zwischen den beiden Definitionen wie folgt zusammenfassen.
In userem Simulation-basierten Modell kann der Prover Informationen über seinen Schlüssel an den Angreifer weitergeben, solange diese Informationen einem Simulator nicht helfen, mit gleicher Wahrscheinlichkeit in der Zukunft zu authentizieren. In der Praxis kann der Prover aber nur sehr wenig Information über seinen Schlüssel an den Angreifer weitergeben, sodass: (i) der Angreifer authentifiziert wird, solange der Prover ihm hilft, und (ii) sobald der Prover ihm nicht mehr hilft, die Wahrscheinlichkeit, dass der Angreifer authentifiziert wird, sehr klein ist. Im Gegensatz dazu ist der Prover im der spielbasierten Modell nicht bereit, *irgendeine* Information über seinen Schlüssel an den Angreifer zu übermitteln, was den Angreifer abschwächt.
Insbesondere erweitern wir vorherige Mafia Fraud Modelle, so dass wir key-learning-Angriffe wie oben beschrieben abdecken können. Solche Angriffe sind besonders nützlich gegen Protokolle, die Resistenz gegen Terrorist Fraud erreichen wollen. Darber hinaus fhren wir eine erweiterte Definition von Mafia fraud ein, welche den Angreifern ermglichen sich auch in solchen Sitzungen zu authentisieren, in welchen sie zu Anfang nur Nachrichten weiterleiten (dies modelliert eine ehrliche Sitzung zwischen dem Prover und dem Verifizierer), und dann versuchen die Angreifer, die Sitzung ohne die Hilfe des Provers zu beenden. Darüber hinaus zeigen wir, dass strMF strikt stärker ist als Resistenz gegen Mafia Fraud.
Wir stellen ebenfalls fest, daß das simulationsbasierte Modell von Resistenz gegen Terrorist Fraud SimTF einerseits sehr

stark ist (wir können zeigen, dass einige sehr prominente Protokolle gemäß unserem Modell unsicher sind), andererseits deckt es aber auch nicht alle möglichen Angriffe ab (da der Prover nur offline mit dem Angreifer interagiert). Daher führen wir sowohl eine stärkere, simulationsbasierte Definition strSimTF ein als auch eine schwächere, spielebasierte Definition GameTFĖrstere gibt noch höhere Sicherheitsgarantien, wohingegen letztere effizientere Konstruktionen erlaubt (wobei sie gleichzeitig offensichliche Angriffe nicht mehr berücksichtigt). We beweisen, dass die bekannten Konstruktionen von Bussard und Bagga [15], die SimTF*un*sicher sind, gemäß GameTF sicher sind. In der Tat erreichen die meisten Protokolle in der Literatur, die Resistenz gegen Terrorist Fraud für sich beanspruchen, das Sicherheitsziel GameTF und nicht SimTFẆir sind der Meinung, dass GameTF zwar ein minimales Maß an Resistenz gegen Terrorist Fraud erreichen, aber sehr einschränkend ist und Angriffe ermöglicht, insbesondere in Angriffsszenarien, bei denen der Prover einige Bits des geheimen Schlüssels freigibt, when der Angreifer dann authentifizieren kann.

Konstruktionen und Compiler. Wir sind die ersten, die (i) ein beweisbar SimTF-sicheres Protokoll entwerfen und zeigen, dass dieses auch strSimTF-sicher ist, und (ii) ein Protokoll entwerden, welches beweisbar sicher Geheimhaltung des Aufenthaltsortes erreicht. Darüber hinaus beschreiben wir Compiler, um Protokolle, welche resistent gegen Mafia Fraud sind, in Protokolle zu transformieren, welche strMFsicher sind. Darber hinaus konstruieren wir Compiler welche Protokolle, die sicher gegen Mafia Fraud sind, in Verfahren transformieren, welche strMFsicher sind. Zustzlich funktioniert unser Compiler selbst dann, wenn das urspngliche Protokoll nicht vollstndig resistent gegen Mafia Fraud Angriffe ist; es darf hingegen durchaus anfllig fr key-learning-Angriffe sein (eine besondere Form der mafia fraud Angriffe). Dieser strMFCompiler benutzt einen finalen lazy Authentisierungsschritt, der Angreifer daran hindert, abgebrochenen Prover-Verifizierer-Sitzungen fortzuführen. Beide Compiler können, mit verschiedenen Optimierungen, auf die meisten Distance-Bounding-Protokolle angewandt werden. Darüber hinaus beschreiben wir verschiedene andere praktische Kniffe, sowohl für die Modellierung (wo wir zeigen, wie man Sicherheitseigenschaften gemäß unserem Modell nachweist), als auch für Konstruktionen (Wir zeigen, dass Impersonifizierungssicherheit leicht zu erreichen ist und wie gegenseitige Authentisierung verwendet werden kann, um rechenlastige Operationen an den Verifizierer outzusourcen).
Zukünftige Arbeit umfasst (i) die Modellierung erweiterter Angriffsszenarien für Distance-Bounding-Protokolle mit mehreren Provern und einem einzigen Verifizierer, oder mehreren Provern und mehreren Verifizierern, was vorige Arbeit von Čapkun et al. [74] formalisieren würde; (ii) die Frage, inwieweit unsere Ergebnisse auf ein Modell ohne Phasen übertragbar sind (so wie zum Beispiel das Modell für Geheimhaltung des Aufenthaltsortes) oder auf ein Szenario mit assymmetrischen Schlüsseln; (iii) die Frage, wie sich die Eigenschaften, die wir definiert haben, zueinander verhalten, zumindest jene Relationen, die noch nicht erforscht wurden wie zum Beispiel die Frage, ob unser Compiler zur Schlüsselaktualisierung GameTF erhält. Darüber hinaus möchten wir betonen, dass verlustlose Sicherheitsreduktionen sehr wünschenswert sind. Es ist daher von Interesse, Compiler zu konstruieren, welche Eigenschaften wir Privatheit, KLMF Sicherheit, und strMF Sicherheit mit verlustlosen Reduktionen erreichen. Eine wichtige, künftige Forschungsrichtung is die Konstruktion von Distance-Bounding-Protokollen für Szenarios mit assymmetrischen Schlüsseln (unsere Ergebnisse betreffen Protokolle mit symmetrischen Schlüsseln). Insbesondere ist die Frage interessant, inwieweit Distance-Bounding auf RFID Tags mit Kryprographie auf elliptischen Kurven angewandt werden kann. Schließlich ist auch eine stabile Implementierung von Distance-Bounding-Protokollen für verschiedene Architekturen ein wichtiges Ziel sowie die Frage, inwieweit diese verwendet werden können, um die in diesem Framework beschriebenen Angriffe zu verhindern.

Contents

1 Introduction

Authentication is a prerequisite of any access control scheme. During authentication, a so-called verifier must either *accept* a prover as being legitimate (when the prover can prove possession of credentials that allow it to authenticate) or reject it if it is illegitimate. Classical applications of authentication range from logistics to public transport, from the Passive Keyless and Start (PKES) systems used in cars to personal identification. Authentication schemes are considered secure if they prevent impersonation attacks, i.e. an illegitimate prover must succeed in authenticating to the verifier with only negligible probability.[1]

However, security models for classical authentication do not capture man-in-the-middle (MITM) relay attacks, where an adversary just forwards data between the prover and the verifier, authenticating with probability 1. Such attacks are called mafia fraud [24]; the adversary in this scenario is typically a coalition of two adversaries, a so-called *leech*, which interacts with the prover, impersonating a verifier, and a so-called *ghost*, which interacts with the verifier, impersonating a prover. The two adversaries usually communicate via fast, reliable communication channels, and, by relaying correct, honestly-generated information between them, they ensure that the ghost, which is an illegitimate party, authenticates to the verifier (thus contradicting our security requirements). This is also depicted in Figure 3. Environments with no central authority and certificates, like RFID identification, are especially subject to mafia fraud, as indicated in [20, 25, 33, 38, 39]; several works also show attacks on the HB protocol [14, 26, 35, 45, 54, 63], which is designed for low-power devices e.g. RFIDs. For an overview of RFID security see [46].

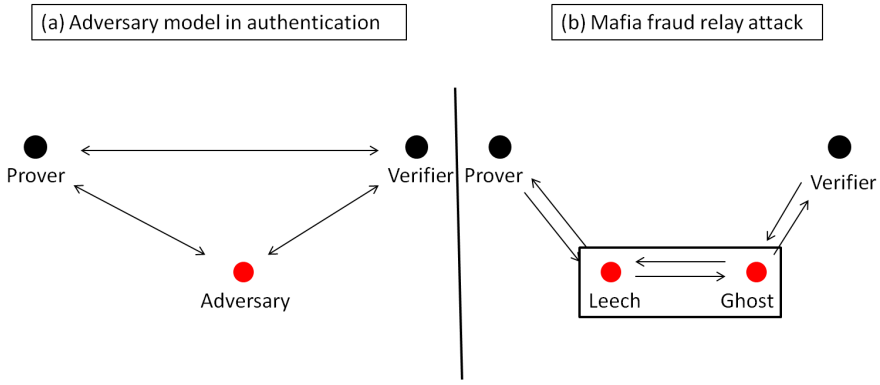

Figure 3: Adversary models in (a) authentication and (b) mafia fraud.

1.1 Distance-Bounding Protocols

Following the idea that relays seem to cause a processing delay in the MITM attacker, Brands and Chaum [13] introduced distance-bounding protocols, where the verifier uses a clock to measure the time elapsed between sending a challenge and receiving the response, thus preventing mafia fraud. The measured time-of-flight is compared to a threshold time $t_{\max}$: if the measurement doesn't exceed $t_{\max}$, then the response is *in time* and assumed to come from a genuine nearby prover. In the original construction due to Brands and Chaum, as well as in many other protocols e.g. [6, 15, 42, 49, 50] the measurements are round-based; each round is either *lazy* (slow) — if the clock is *not* used — or *time-critical* (fast) — if the clock measures time-of-flight.

Initially, the three main attacks distance-bounding protocols had to address were:

1. Mafia fraud. Here the adversary runs a MITM interaction with an honest prover and an honest verifier, without purely relaying messages (since we assume the verifier's clock detects pure relay). The adversary's aim is to be authenticated by the honest verifier.
2. Terrorist fraud. The terrorist adversary now collaborates with the prover (which is dishonest) in an offline fashion (i.e. the adversary does not query the prover in rounds where the clock is used). The prover shouldn't, however, give away trivial information to the adversary, e.g. the secret key.

[1] In identification schemes, where the verifier also establishes the exact identity of the prover as well as its legitimacy, impersonation resistance also entails that no legitimate prover can impersonate another prover.

3. Distance fraud. A distance fraud adversary is a dishonest prover, whose position is outside the range associated with $t_{\max}$ from the verifier. Now the adversary's goal is to authenticate to the verifier, thus fooling the verifier's clock.

Subsequently, Avoine and Tchamkerten [6] also suggested a fourth attack, namely slow-round impersonation resistance. This attack is especially applicable to cases where distance-bounding protocols are implemented on resource-constrained devices, e.g. RFID tags, which cannot support many time-critical rounds. Since in many distance-bounding protocols the level of impersonation security heavily depends on the number of time-critical rounds, the overall security level for resource-constrained implementations would then become too low in practice. By employing lazy-phase authentication, the security level of the protocol is increased.

We exemplify the first three attacks as follows (see [27]): let Alice hold the unique pass key (which acts as a prover) to a gym locker, equipped with the corresponding verifier (which has a clock as described above). One evening, Alice is not at the gym, but at a party. In the *mafia fraud* scenario, Bob *is* at the gym while his accomplice, Bobette, is at the party with Alice (in mafia fraud terminology, Bob is the *ghost*, while Bobette is the *leech*). Bob wants to open the locker (without Alice's consent for mafia fraud). In this attack, Bob and Bobette relay messages between the locker and Alice's pass-key such that Bob gains admission (note that the verifier's clock will prevent pure relaying of communication).

However, if Alice and Bob are friends, Alice may *agree to let* Bob use her locker (for this night only). This is the *terrorist fraud* scenario, where Alice may forward Bob information allowing him to use her locker. However, Alice doesn't want Bob to abuse her kindness and open the locker on his own, this or any other time. For terrorist attacks thus, Alice helps Bob herself (in an offline fashion, without providing assistance during the time-critical phases): Bobette is not needed.

Finally, if Alice parked her car in a bad spot, she might want to "prove" that she was at the gym instead by opening her locker. This is the *distance fraud* scenario, where Alice is the adversary. Since she holds the unique pass key to the locker, Alice will be able to prove that she was at the gym if the verifier accepts her authentication despite the fact that she is *not* at the gym.

1.2 Prior and Related Work

Since their introduction, distance-bounding protocols have generated several important research directions, which can be roughly divided in the following three categories: (i) attack implementation; (ii) protocol design; (iii) security models. We discuss the progress (prior to, or concurrent with, our contributions) in each of these areas below.

Attack Implementation. The earliest — to our knowledge — mafia fraud impersonation attacks date back to 2004, when Levi et al. [55] showed how to use mafia fraud on Bluetooth devices, noting that such devices would be completely unaware of a MITM adversary relaying messages between them. In 2005, Kfir and Wool [48] showed how to implement mafia fraud attacks against smartcards, whereas Hancke implemented a practical attack against RFID tags, i.e. ISO standard 14443. This result was significant in particular because RFID had become popular, cost-efficient implementation platforms especially for lightweight authentication protocols, e.g. HB and its variants [14, 26, 35, 45, 54, 63]. Hancke generalized his attack for all proximity protocols in [40], and during the following year, practical mafia fraud attacks were also noted in the context of e-passports [44]. Recently, several other attacks were shown against RFID-based e-voting schemes [62] and passive keyless entry and start (PKES) systems in cars [33]. A mafia fraud attack implementation where adversaries use NFC phones was shown by [34]. Thus, practical implementations suggest that mafia fraud attacks are feasible in reality, which is a strong incentive to design and implement authentication by means of distance-bounding protocols which are provably resistant to the attacks outlined above.

Protocol Design. Several existing protocols implement resistance against one (or more) of the above threats. To name just a few constructions, we have the initial schemes due to Brands and Chaum [13], the well-known protocol due to Hancke and Kuhn [42], the protocol due to Avoine and Tchamkerten [6], and the construction of [49]. These schemes claim mafia and distance fraud resistance levels using a metric called the False Acceptance Rate (FAR), which measures the probability that an adversary succeeds in an impersonation attempt against an honest verifier. For most constructions, security is achieved during the time-critical rounds, which are more or less independent. Since the prover and verifier exchange just bits during the time-critical rounds, the best security one can achieve per round is $\frac{1}{2}$; this adds up to a total security level of 2^{-N_c} for N_c time-critical rounds. Whereas the protocol due to Brands and Chaum achieves this security level, the subsequent constructions of Hancke and Kuhn and resp. Avoine and Tchamkerten do not, since a MITM adversary can query the honest prover in advance to learn some of the correct responses. The direct consequence is that one needs more time-critical rounds in comparison to the Brands and Chaum protocol to achieve the same security level. An improvement introduced by Kim and Avoine is partial verifier authentication, which partly prevents this attack, and thus comes closer to the $\frac{1}{2}$ time-critical-round security mark. The claimed security levels of these schemes are summarised in Figure 4, where

we indicate, under "Rounds", the number N_c of time-critical rounds required for a mafia resistance of about 2^{-k}. We round down the number of rounds in [42] to $2k$.
There are fewer protocols in the literature that address terrorist fraud resistance; amongst these we mention the constructions in [15, 50, 70]. The idea introduced by Bussard and Bagga is to thwart terrorist attacks by relating time-critical responses to a long-term secret, such that by aiding the adversary, the prover must reveal this long-term secret. This same idea is used later by constructions such as [70] and [50]. We also show the claimed security levels of a selection of these protocols also in Figure 4, where we also round down the number of time-critical rounds in [70] to $2k$. We include the following constructions: Brands and Chaum [13], Hancke and Kuhn [42], Avoine and Tchamkerten [6], Reid et al. [70], and Kim and Avoine [49].

	[13]	**[42]**	**[6]**	**[70]**	**[49]**
Mafia	✓	✓	✓	✓	✓
Terror	×	×	×	$(\checkmark)^1$	×
Distance	✓	✓	✓	✓	✓
Impersonation	×	×	✓	×	×
Rounds N_c	k	$> 2k$	k	$> 2k$	k
Storage	N_c	$2N_c$	$O(2^{N_c})$	$2N_c$	$4N_c$
Private-key	×	✓	✓	✓	✓

Figure 4: Claimed Security and Actual Efficiency of Distance Bounding Protocols at a glance ([1]only special terrorists, no formal proof)

We note that the security proofs of these schemes were done in the respective papers by using the FAR. However, the false-acceptance rate is only an intuitive metric, in the absence of a formal framework, and it may enable practical attacks against distance-bounding schemes. In fact, [1] recently shows attacks against two distance-bounding authentication schemes, namely the Hitomi and NUS protocols.
Also note that distance-bounding protocols were recently implemented by Rasmussen and Čapkun. [68] and Ranganathan et al. [66], who proposed the first practical implementations of such protocols by using analogue components, which allows for the necessary small processing delay.

Security Models. Though distance-bounding protocols abound in the literature, their security was not formalized until very recently, when Avoine et al. [4] introduced a first — and quite informal — model for security in distance-bounding protocols. In general lines, this framework accounts for the more common MITM strategies in mafia and terrorist fraud, as well as for distance fraud scenarios, sketching their security notions in terms of these basic attack strategies. They consider both adversaries where the protocol is used as a black box *and* scenarios where the prover may tamper with the protocol run during e.g. distance-fraud attacks. We refer in more detail to this framework in Section 2.2.2.

1.3 Contributions and Roadmap

Our results fall chiefly under the categories of security models and protocol design. Though we discuss each of our contributions more in detail in Chapter 7, we also briefly review them here, indicating their significance.

1.3.1 Models

Fundamental threats. We consider distance-bounding protocols from three different perspectives, which we view as tiers of our formal framework, as depicted in Figure 5. This figure is also shown in Chapter 7. The central layer of this framework, presented in detail in Chapter 2, formally defines the following four notions: (i) mafia fraud; (ii) terrorist fraud; (iii) distance fraud; (iv) offline impersonation security. The exact security notions we define enable the adversary to choose its strategy arbitrarily, with as few restrictions as possible, which makes our definitions very strong. We also show how to attain these notions, showing that they are also not *too* strong. Furthermore, we show that, contrary to the remarks of [4] and [70], the four notions are independent: in fact, we are able to describe protocols that have three of the four properties, but are vulnerable to the remaining attack.
Furthermore, we assess the exact (rather than asymptotic) security of a few prominent constructions in the literature in our framework, namely the protocols due to Brands and Chaum [13], Hancke and Kuhn [42], Avoine and Tchamkerten [6],

Reid et al. [70], the Swiss-Knife protocol of [50], and an improved version of the protocol due to Kim and Avoine [49]. In Chapter 4, we also assess the security of the Bussard and Bagga protocol [15]. Our analysis shows that, contrary to the claims of [15, 50, 70], the protocols suggested in each of these papers are *not* terrorist fraud resistant in the sense of our simulation-based definition. In Chapter 5 we show that these protocols attain a different notion of terrorist fraud resistance.

Serge Vaudenay pointed out [11] that many of the protocols we discuss in this work rely incorrectly on the security of a pseudo-random function (PRF) towards proving distance fraud resistance. In order to bypass the flaw he outlined, we slightly modify most distance-bounding protocols in the literature, thus ensuring that the prover cannot manipulate the PRF in order to obtain an output with low entropy.

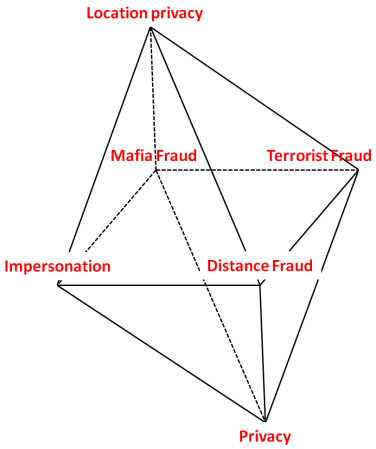

Figure 5: Model in a nutshell

Privacy in authentication. In the bottom tier of our model, we consider privacy in authentication, as defined by [73]. As we explain more in detail in Chapter 3, Vaudenay's model considers both narrow adversaries (who cannot see the bit output by the verifier at the end of the protocol run) and wide adversaries (who do see whether the prover authenticates or not). There are four main degrees of privacy: weak, forward, destructive, and strong privacy. Our view is that distance-bounding protocols are an extended version of authentication, thus we use the results of Vaudenay [73] to both define and attain narrow-destructive privacy in our symmetric-key setting.

Towards this goal, we note that the model we present in Chapter 2, covering the central tier of our framework, only covers distance-bounding attacks for static keys, i.e. the authentication key associated with the honest prover is not updated. However, in order to achieve privacy, key updates are necessary. As the adversary can now try to run denial-of-service (DoS) attacks, such that the honest prover is no longer able to authenticate to the verifier, we introduce a further property, (v) availability, which is a long-term completeness, requiring that an adversary is unable to desynchronise the prover and verifier state such that the prover is no longer able to authenticate. The other notions (i.e. properties (i) to (iv)) carry over naturally from the static setting to the key-update setting. We show a compiler that takes as input a mafia, distance, and impersonation secure construction in the static-key setting and outputs a protocol that preserves the same levels of security (up to negligible factors) in the presence of key update, which is furthermore available in the sense of property (v). Furthermore, if the input protocol is weak-private in the sense of Vaudenay [73], then the scheme output by the compiler is also narrow-destructive private in the definition of [73]. This ensures that distance-bounding protocols achieving a very weak notion of privacy are sufficient in practice, since running our compiler on the schemes will automatically enhance their privacy, while preserving their distance-bounding properties. We note that this is the best we can hope for, since narrow-strong privacy requires key agreement [73].

We describe this compiler in more detail under Protocol Design, below.

Location privacy. Finally, the upper tier features location privacy, a notion considered for the first time in the context of distance bounding by Rasmussen and Čapkun [69]. However, this paper does not actually model the idea of location privacy, merely showing a protocol that claims to be location-private. Unlike most other distance-bounding protocols,

this construction is *not* round based: in fact, in order to achieve location privacy, the prover and verifier in this protocol communicate simultaneously, transmitting bit-streams. In Chapter 6 we define location privacy as the property that nothing can be learned about a prover's position apart from the fact that it is within a verifier's proximity. Thus, we are able to prove that the protocol due to Rasmussen and Čapkun does *not* preserve location privacy. In fact, our result is much stronger: we show that, in the absence of very large delays (exponential in the security parameter), location privacy cannot be achieved even computationally, and even if the adversary is limited, i.e. it cannot gauge signal strength or use triangulation to locate a signal's source. Furthermore, we also show a mafia fraud attack against the RČ protocol.
Our approach is as follows. We begin by changing the communication model to account for continuous transmissions. In this new model, the prover and verifier interact with each other by using two duplex channels: a *timeless* channel, where parties simply exchange messages; and a *timed* channel, where parties send messages at certain transmission times, and each party receives the message at a time associated with the communication distance between itself and the sending party. In other words, the adversary receives a transmitted bit, with a delay corresponding to its distance to the sender, with respect to the sending time. Sending and receiving times are considered bit-by-bit, so as to better capture reality.
Furthermore, noting that the usual definition of location privacy cannot be achieved for distance-bounding (since distance-bounding protocols always reveal *some* information about the prover's location, namely whether or not the prover is within a range corresponding to $t_{\max}$ from the verifier), we define location privacy for distance-bounding protocols in terms of a left-or-right indistinguishability game. An adversary wins this game if it can link distance-bounding sessions with tags sharing the same secret key, but placed in different positions. We show that location privacy can never be achieved in the presence of powerful, so-called omniscient, adversaries, which are able to learn the sending time of messages sent on the timed channel (e.g. by triangulation or measuring the signal strength of honest transmissions). Furthermore, achieving location privacy even against weaker, limited adversaries seems unfeasible, as it introduces a large computational overhead. Namely, we show that the success probability of such a weak adversary is non-negligible, unless the prover and verifier introduce significant delays in their timed-channel communication. As we also briefly describe in Section 1.3.2, we also show a scheme that achieves location privacy in the presence of weaker adversaries: this scheme relies on a modified version of the RČ distance-bounding protocol, which also prevents a particular mafia fraud attack.
We also show this attack in detail in Chapter 6. The main idea is that the prover chooses its response nonce at the beginning of the protocol execution, and sends this value (encrypted and signed) to the verifier. Even if the signature scheme is chosen so that it hides the message, an adversary can replay the values in a fresh authentication session; by guessing when the prover started responding by using the nonce, the adversary also manages to respond correctly in a fresh authentication phase, thus authenticating to the verifier.

Further insight. Zooming in on the central tier of our model, we also investigate mafia and terrorist fraud resistance in more detail. Our main result is, on the one hand, to capture terrorist fraud attacks in a much more complete fashion, and on the other hand, to incorporate mafia fraud attacks in the presence of aborts. We illustrate the gist of our results in Figure 6, which also appears in Chapter 7.

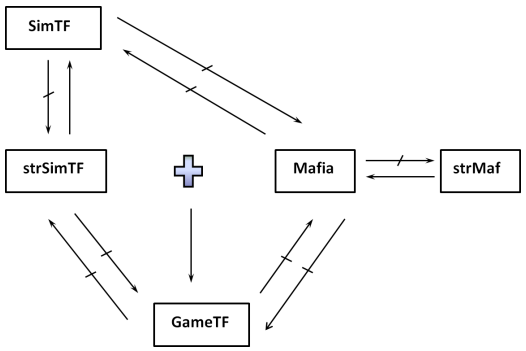

Figure 6: Flavours of mafia and terrorist fraud.

Terrorist fraud. In the context of terrorist fraud, our extended model centrally revolves around the amount (and format) of assistance the prover gives to the adversary. We consider that, depending on the incentive behind the attack, a dishonest prover might consider it acceptable that *some* of the information forwarded to the adversary can subsequently help this

adversary authenticate, just as long as it doesn't give him an equal probability to authenticate as it has *with* the prover's aid. We consider the problem of terrorist fraud resistance in depth in Chapter 5. This notion is controversial in the literature, as it seems hard to capture what is generally meant by the dishonest prover forwarding the adversary trivial information.
Our initial terrorist fraud definition in Chapter 2 is simulation-based, very precise, but also very strong. In what follows, we call this notion SimTF security. In Chapter 5, we show two alternative definitions of this notion, one stronger, capturing the fact that the dishonest prover may also help the adversary offline, and another, independent notion, which captures the intuition that the dishonest prover is unwilling to help the adversary, if this help lets the adversary gain even a small advantage in subsequent authentication sessions.
The idea behind SimTF security is that if the adversary (aided offline by the prover) authenticates with some probability, then a simulator can (on its own) recover enough information from the adversary's state to authenticate *with the same probability*. Thus a protocol is terrorist fraud resistant if an adversary only wins upon receiving non-trivial information from the prover, like a long-term secret key. Note that in this scenario, the dishonest prover is willing to impart *some* significant information to the adversary, as long as this information doesn't allow the simulator to authenticate with *equal* probability.
In Chapter 2 we prove that some schemes claiming to be terrorist fraud resistant are not, in fact, provably SimTF secure. Indeed, designing such schemes is tricky, as finding a simulator is challenging. The definition is very strong since it only excludes attacks where (parts of) the secret key are *directly* given to the adversary (enabling the simulator to use them as effectively). In Chapter 5 we first show that SimTF security is, in fact, achievable by describing the first SimTF secure protocol, which we describe in more detail in Section 1.3.2. Our protocol in fact attains a stronger notion of terrorist fraud resistance, which we denote strSimTF security. This model closely resembles SimTF security; however, the prover can now also provide online assistance to the adversary.
Most schemes in the literature do, in fact, attain some form of terrorist fraud resistance, which we call game-based terrorist fraud resistance GameTF. This notion captures the idea that the prover is not willing to forward *any* information about its secret key to the adversary. Thus, in GameTF we exclude attacks where the prover's help facilitates the adversary's ulterior attempts (without the need of a simulator who must win with the same probability). This model is independent of the approach of Chapter 2 but captures the intuition of terrorist fraud. We prove the protocol of Bussard and Bagga [15] GameTF secure, thus showing that GameTF security enables more efficient constructions[2].
Finally, we also relate these flavours of terrorist fraud resistance and show separations between them. Interestingly, the strSimTF and GameTF models turn out to be independent; however, a protocol achieving strSimTF security *and* mafia fraud resistance is also GameTF secure. We also show that, though our GameTF definition resembles the informal notion in [4], it does not imply mafia fraud resistance (contrary to the results of [4]). In fact, we show a full map of the relations in Chapter 5.

Mafia fraud. In the case of mafia fraud resistance, we argue that some distance-bounding protocols in the literature, namely those claiming to achieve terrorist fraud resistance, are vulnerable to an attack known in authentication, but not explicitly considered in distance-bounding: key-learning attacks. In particular, such attacks are not considered as part of the previous mafia fraud model of Avoine et al. [4].
Key-learning attacks exploit related time-critical responses, tied to a long-term secret. The gist of the attack, explained informally, is as follows: an attacker first mostly observes a few prover-verifier protocol runs, flipping one or a few challenge bits, such that the adversary learns the long-term secret bit by bit. Finally, once the key is recovered, the attacker can use it in order to authenticate to the verifier. In the running example of Alice and Bob, key-learning attacks occur when Bob tries to flip some bits during one of Alice's genuine authentication attempts, not with the purpose of authenticating, but with the purpose of recovering Alice's key. In Chapter 4, we discuss such attacks in more detail.
We also consider the issue of aborts, i.e. the case where the prover begins an authentication session, but then stops contributing to the session. In this case, an adversary in proximity of the verifier can hijack the sessions and take over from the prover, thus being authenticated by the verifier. Thus, if Alice is e.g. called away from the locker just as she started the authentication, we stipulate that Bob should not be able to just "take over" from Alice and authenticate in her stead. In this case, we define strong mafia fraud attacks (strMF), where the adversary's goal is to authenticate to the verifier in a verifier-adversary session where either less than the allowed number of phases were relayed, or, if more phases are relayed, there exists a phase (time-critical or lazy) in which the prover stops communicating and the adversary must forward correct responses on its own. We also show how to turn a mafia fraud resistant construction into a strong mafia fraud resistant protocol (by using a final authentication response from the prover to the verifier, such that it cannot be replayed); furthermore, we show that the same compiler can be used to attain strong mafia fraud resistance from something

[2]Note that the scheme in [15] is vulnerable to the same attack that we describe against the protocol of Reid et al. [70] in Chapter 2. Thus, the Bussard and Bagga protocol is SimTF and strSimTF insecure.

less than mafia fraud resistance: indeed, the compiler also works if the underlying scheme is vulnerable to key-learning attacks, but not to any other mafia fraud attacks. We briefly discuss this compiler below.

1.3.2 Protocol Design

SimTF **security.** Our results in protocol design follow three main directions: constructions, attacks, and compilers. On the one hand, we show how to attain SimTF security, which is a strong, simulation-based version of terrorist fraud resistance. None of the protocols claiming to achieve terrorist fraud resistance in the literature do, in fact, attain the SimTF notion: the difficulty is that SimTF security proofs require the construction of a simulator that authenticates with the same probability as the successful adversary. In fact we show a generic attack against the SimTF security of a particular type of protocols in the literature claiming to achieve terrorist fraud resistance. Our SimTF secure scheme bypasses this attack by introducing a back door, allowing the simulator to authenticate if it can reconstruct (up to a few bits) the long-term secret. This construction actually also attains the strong security notion of strSimTF security, where the adversary is also allowed to receive online support from the dishonest prover.

Location privacy. A second construction is a location-private protocol, relying on some ideas introduced by Rasmussen and Čapkun [69]. We first modify the protocol in order to thwart a mafia fraud attack (also outlined in this thesis), and then use our result, indicating that an exponential time-delay is required to achieve location privacy, in order to show a computationally location-private distance-bounding protocol. The mafia fraud attack against the scheme in [69] relies on the fact that the prover (i) chooses its authentication response independently of the verifier; and (ii) commits to this response at the start of the protocol. In our attack, the adversary replays the commitment and tries to guess when to send the de-committed response. Furthermore, in order to achieve location privacy, we also introduce a delay in the communication.

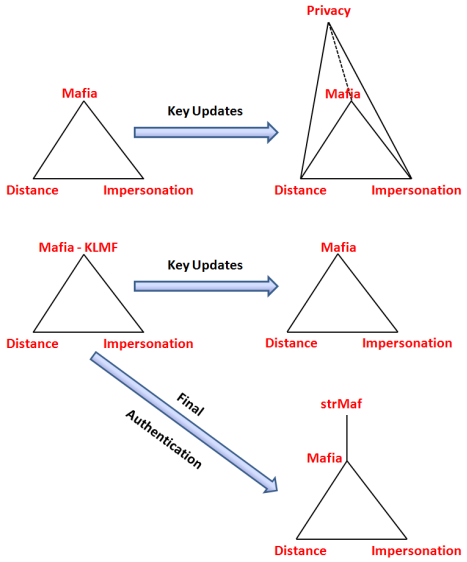

Figure 7: Compilers in this thesis. The pyramid on the upper row refers back to our three-tiered model, showing that the first compiler transforms protocols in the central tier of the model to protocols in the privacy tier. The two-dimensional depictions in the lower rows refer to the in-depth treatment of mafia fraud resistance, which fit into the main tier of the framework. Here, (Mafia - KLMF) denotes that the protocol is not resistant to key-learning attacks, but it is resistant to any other kind of mafia fraud attack.

Compilers. Finally, we also show two compilers, which are related as in Figure 7. Compilers are very important in cryptography, because they allow protocol designers to aim for protocols attaining weaker properties (since the compiler can then be applied to the obtained schemes, enhancing their properties). Thus, one can simply construct protocols which are mafia, distance, and impersonation secure in the static key model, and our key-updates compiler will enhance

their privacy from narrow-weak to narrow-destructive privacy. Similarly, one can construct protocols which are mafia fraud resistant and then apply our compilers to achieve resistance against key-learning attacks (see the Models paragraph above). We show two compilers: one for key updates, and another achieving mafia fraud security in the case of aborts. These compilers relate to each other as in Figure 7, and we briefly describe them below.

The main idea of our key update compiler compiler is to let the prover update state early, after an initial lazy-phase authentication step, and to make the verifier update state only if both this authentication step and the distance-bounding authentication in the underlying protocol succeed. Thus, the prover updates state more often than the verifier; however, during the initial authentication phase, the verifier is able to "catch up" with the prover's state. Furthermore, in order to ensure that the adversary cannot make the verifier enter an endless iterative loop, we make the prover keep two state fragments, one which is always updated, and the other which enables the verifier to only check the authentication string against only two possible values for each prover. More details can be seen in Chapter 3.

In order to achieve the strong mafia fraud property (i.e. mafia fraud resistance in the presence of aborts), we use a trick also used by Kim et al. [50] in the Swiss-Knife protocol, namely we add a final round of lazy authentication between prover and verifier. This ensures that an adversary hijacking a communication attempt must in fact succeed in this last authentication round. We show that in fact the compiler also works if the protocol is not entirely mafia fraud resistant, but rather is vulnerable to key-learning attacks. We argue that the same type of protocols can be made mafia fraud resistant if we use e.g. key updates.

2 The Distance-Bounding Framework

This chapter introduces a formal model for assessing the security of distance-bounding authentication protocols. Though our framework (published in [27]) is general, we also make special provisions for parameters occurring naturally in RFID authentication, where the communication is unreliable and often the RFID tag's communication capacity is restricted. Hence we also often use RFID-specific terminology, referring to provers as tags and to verifiers as readers. We discuss our definitions in view of related work, with particular focus on the recent attempt to formalise distance bounding security due to Avoine et al. [4]. This work is published in [30]. Furthermore, we discuss some practical and implementation-related notions related to distance-bounding protocols in general, work which appears in [32]. We note that a particularly important modification was made to [27] and [32] as a consequence of the work of Boureanu et al. [11]. This modification concerns distance-fraud resistance vulnerabilities of various RFID protocols, due to pseudo-random function (PRF) specifics, and its most significant effect is that most distance-bounding protocols to date are not provably distance-fraud resistant.
After introducing the framework we also give a security analysis of the following prominent constructions in the literature (in the order in which we handle them).

- The original distance-bounding protocol due to Brands and Chaum [13].
- The construction due to Hancke and Kuhn [42].
- The protocol with dependent challenges due to Avoine and Tchamkerten [6].
- An improved version of the protocol due to Kim and Avoine [49].
- The terrorist fraud resistant construction due to Reid et al. [70].
- The Swiss Knife Protocol due to Kim et al. [49].

Contributions. In short, this chapter could be divided into three main sub-topics: (1) the framework itself; (2) its relation with previous literature, particularly the framework of Avoine et al. [4]; and (3) the formal assessment of various existing distance-bounding constructions in the literature. We briefly outline the most important contributions within each topic.

1. The Framework. The main contributions in this area could be summarised as follows:
   - We give rigorous cryptographic security models for mafia, terrorist, and distance fraud, and impersonation security.
   - Thus, we are able to relate the security properties formally. We also refute the claim in [70] that terrorist fraud resistance implies distance fraud resistance.
   - We also review some particularities of RFID hardware and communication, explaining why it is vital to consider inconsistencies in communication when we speak of RFID distance bounding.
2. Related work. Though we mainly focus on the previous distance-bounding framework by Avoine et al. [4], we also refer to other related work, such as the recent work on (quantum) position-based cryptography due to Chandran et al. [18], self-delegatable schemes [36], and the transferability of anonymous credentials, as defined by Camenisch and Lysyanskaya [16].
3. Protocol Analysis. We also provide a comprehensive analysis of several protocols in the literature, namely: Brands and Chaum [13], Hancke and Kuhn [42], Avoine and Tchamkerten [6], Reid et al. [70], the Swiss-Knife protocol [50], and an improvement of the scheme due to Kim and Avoine [49]. In particular, we:
   - Show concrete security bounds for the mafia fraud, terrorist fraud, distance fraud, and impersonation resistance for each of these protocols.
   - Disprove claims of terrorist fraud resistance of the well-known protocols in [70] and [50].
   - Show that the generic attack of Boureanu et al. [11] applies to a wide variety of protocols, which are thus vulnerable to distance fraud attacks.
   - Show that the protocol of Reid et al. [70] is susceptible to a key-learning mafia fraud attack if the symmetric encryption function used by the protocol is instantiated as bitwise XOR. For this attack, an adversary is able to learn a long-term secret key in a bit-wise manner, by taking advantage of the inter-dependency of time-critical responses.

- Discuss the consequences of the contradiction between the properties claimed by the authors of both schemes, and the properties we can prove they achieve. In particular, we balance the strength of terrorist fraud resistance against the notion we intuitively wish to achieve.
- Outline a few strategies for distance-bounding constructions.

2.1 Security in Distance-Bounding Protocols

In general, distance-bounding protocols are an enhancement of standard authentication between provers and verifiers. In authentication, the protocol yields as a result a verifier-generated bit, which denotes either that (a) the verifier accepted the prover, i.e. the prover has proved its legitimacy; or that (b) the verifier rejected the prover, i.e. the prover was illegitimate. In distance-bounding protocols, the verifier, equipped with a clock, also rejects if the prover is not within a certain "trusted" distance (measured in time-of-flight) from the verifier. Throughout this thesis, we often call this trusted distance "proximity", and say that a prover is within the verifier's proximity if its distance between the prover and the verifier is less than a given, pre-set threshold

Though distance-bounding protocols can be run on many distinct types of hardware, it is more challenging to implement such protocols on resource-constrained hardware, such as RFID tags. On the other hand, however, RFID hardware is cost-efficient and largely deployed in systems where authentication is required, such as logistics, public transport, etc. Thus it is also interesting to consider distance-bounding on RFID hardware, where one must take into account a few peculiarities of RFID computation and communication.

This section is structured as follows: we first start by reviewing practical aspects of distance bounding at large and RFID distance-bounding authentication in particular. Then we discuss our contributions in more detail, giving an overview of our models. The actual description of our framework begins in Section 2.1.1, where we describe the attack model. Then we proceed in Section 2.1.2 to give concrete definitions for the four security threats: mafia fraud, terrorist fraud, distance fraud, and lazy-phase impersonation. Finally, we relate the models in Section 2.1.3.

The Practice behind the Theory. Practical investigations [17, 20, 41, 42, 67, 70] indicate some design issues for RFID distance-bounding protocols. As such considerations apply for all low-power devices, we provide for them in our framework.

Bandwidth. Time measurements are very fragile when the parties send other messages than bits [20, 41, 70]. The reason is that on the one hand, fresh noise is introduced in the communication, and on the other hand, the unreliability of the transmission increases with the size of the transmitted message. Distance-bounding protocols should thus only exchange bits for time-critical steps (we still formulate models for arbitrary transmissions).

Computations. Fast-phase computations should be very simple, otherwise they dominate the round-time. It is also important that the computation-time for each round is constant.

Storage. To be suitable for low-power devices, RFID distance-bounding protocols must require only little storage.

Noise. Both transmissions and time measurements are subject to noise [20, 41, 70]. Considering, as above, that only bits are exchanged in fast phases (where the verifier measures the roundtrip time-of-flight), immediate error correction for example is impossible. Hence, our model and protocols take into account threshold levels for failures during timed phases: concretely, the verifier will allow the prover to respond late in a number of rounds. The models are thus more profound, as the adversary can now run the man-in-the-middle (MITM) strategy for those phases.

Early bit detection. Depending on the physical implementations the adversary may be able to predict a transmitted bit "halfway through the signal" [25], allowing the adversary to be fast in its attack. Also, the computation time of the parties in a fast, clocked phase may depend on the actual value received. In other words, the adversary may occasionally be able to derive information from the reader or tag faster than expected. Hence, our model allows the adversary to relay information as long as it is not purely duplicated.

Offline Authentication. Distance bounding is often achieved by timing the communication between the reader and the tag; this is implemented in a few fast communication rounds [6]. Due to hardware constraints, one cannot use many such rounds, and therefore further authentication should be used. As pointed out in [6] it is preferable that the basic authentication step be done before the fast phase begins. Some protocols do not have this property [13, 42], whereas we give a strong definition of it and suggest it as an enhancement of solutions such as [49].

The Models. A sound modelling of feasible attacks is crucial to assessing protocol security. Confusions appear especially with attack modes and successful man-in-the-middle (MITM) attacks, e.g. for the HB protocol [45, 35, 26, 14, 63, 54]. As another example, the allegedly-secure Hitomi and NUS protocols were recently proved insecure [1]. Our models follow game-based approaches, but we also consider practical conditions. This enables us to formally prove that, contrary to the remark of [70], terrorist fraud resistance does not imply distance fraud resistance. In fact, we show that mafia fraud resistance, terrorist fraud resistance, and distance fraud resistance are all independent. More precisely, we present protocols that are vulnerable to one attack, but resistant to all others (including the basic authentication-protocol requirement of impersonation security). Thus, in particular, terrorist fraud resistance does not imply mafia fraud resistance, nor vice versa. Some groundwork has already been laid in this field by Avoine et al. [4], who model mafia, distance, and terrorist fraud in both a white-box sense (i.e. giving adversaries access to the implementation of the prover's protocol) and a black-box sense (adversaries run provers as black boxes). In the framework of [4], distance-bounding protocols have two main goals: authentication and distance checking; each type of fraud is also more formally defined. Adversaries in this framework may choose from three main strategies: pre-ask (query prover before being queried by verifier), post-ask (query prover after being queried by verifier), and early-reply (respond before verifier sends query, without querying the prover). In the black-box model, mafia and terrorist fraud are proved equivalent, whereas terrorist fraud resistance is said to imply distance fraud resistance. In the white-box model, Avoine et al. [4] proves that terrorist fraud resistance implies both mafia and distance fraud. Mafia fraud resistance is equivalent in the black and white-box models, and white-box terrorist and distance fraud are strictly stronger than the black-box notions. Our results are very different, since our models differ subtly from [4]; we review and compare our work in detail to [4] in section 2.2.2. We also relate the notions we define in Section 2.1.3.
Our definitions are much more concrete and formal than those of Avoine et al. [4]. Furthermore, we also include an attack that is not accounted for by previous models, and which we call key-learning attacks (see Section 2.3.5 and Chapter 4). Protocols have many rounds (lazy or time-critical), and adversaries choose (possibly different) strategies for each round, unlike [4]. Our mafia adversaries may relay *parts of the communication*, e.g. flip bits, or purely relay (taint) some rounds. Our mafia and terrorist provers may be anywhere, unlike [4], where provers are outside the target distance from the verifier. By using a simulator, we concretely define "advantage for future attacks" [4] for terrorist fraud. Hence, we prove that all security notions are independent. We also extend impersonation resistance to lazy-phase authentication, thus preventing information leaks to fake provers.

2.1.1 Preliminaries

We consider a single verifier $\mathcal{V}$ (in this chapter we in fact refer to verifiers as readers, as in the RFID scenario) and a single prover $\mathcal{P}$ (an RFID tag), sharing a secret key generated through a key-generation algorithm Kg. To the reader we associate a clock and a database entry storing the tag's secret key. We assume that the authentication scheme $\mathsf{DB} = (\mathsf{Kg}, \mathcal{P}, \mathcal{V})$ marks (consecutive) steps of the authentication protocol as *lazy* or *time-critical*: in time-critical steps, one party —usually the reader— measures the round-time Δt and compares it to a predetermined threshold $t_{\max}$; else the phase is called lazy. A protocol run can consist of arbitrary *non-overlapping* sequences of lazy and time-critical phases, with time-critical phases possibly following one another. Denote by N_c the number of time-critical phases. Errors due to time-measurement noise are modelled by allowing $T_{\max}$-many round-times to exceed $t_{\max}$. Similarly, $E_{\max}$ is the maximum number of time-critical phases with erroneous transmissions.

Definition 2.1 *An* authentication scheme for timing parameters $(t_{\max}, T_{\max}, E_{\max}, N_c)$ *is a triplet of efficient algorithms* $\mathsf{DB} = (\mathsf{Kg}, \mathcal{P}, \mathcal{V})$ *with:*

KEY GENERATION. *For parameter* $n \in \mathbb{N}$, Kg *generates a secret key* sk.

AUTHENTICATION. *The joint execution of algorithms* $\mathcal{P}(sk)$ *and* $\mathcal{V}(sk)$ *generates, depending on* $t_{\max}$, $T_{\max}$, $E_{\max}$, N_c, *a verifier output* $b \in \{0, 1\}$.

We assume that the scheme is complete: for any $n \in \mathbb{N}$ *and any key* $sk \leftarrow \mathsf{Kg}(1^n)$, *the decision bit* b *produced by honest party* $\mathcal{V}(sk)$ *interacting with honest party* $\mathcal{P}(sk)$ *under the requirements following from the timing parameters, is* 1 *with probability (negligibly close to)* 1.

Note that although most definitions of distance-bounding protocols omit $t_{\max}$, this parameter is a crucial difference between distance-bounding and common authentication protocols; alternatively, we could assume that in authentication protocols, $t_{\max}$ is by default set to be infinitely large (the reader accepts tag responses no matter how late they arrive). The parameters $E_{\max}$ and $T_{\max}$ are intrinsic to communication over noisy channels (e.g. RF channels between readers and passive and

semi-passive RFID tags[3]). In distance bounding, it is unreasonable to separate the *reliability* of the communication from its *security*; these properties are connected by the importance of round-time measurements towards acceptance or rejection. Bit errors are unavoidable in RF communication, as stated in point 4 of Clulow et al.'s principles for secure time-of-flight distance bounding [20]. Thus, RF communication noise implies that transmissions between readers and tags are not always reliable, possibly reaching the reader outside the time bound. We can, however, set $T_{\max} = E_{\max} = 0$ for extremely reliable scenarios.

It is also easy to extend the above definition to capture RFID systems relying on some shared public information, where readers use public-key schemes. We can add an algorithm GenPar taking as input a security parameter and generating a master secret/public key pair (K_S, K_P), such that K_S is given to the reader and the key generation algorithm Kg for the tags takes K_P. Completeness then requires that the honest verifier accepts honest provers for all pairs (K_S, K_P), if K_P is used to generate the tag's secret parameters.

2.1.2 The Model

The Communication Model. We use game-based definitions, and we give an adversary $\mathcal{A}$ access to: a reader instance to which it impersonates the tag (a *reader-adversary session*), a tag instance to which it impersonates the reader (*adversary-tag session*), and an interface observing a genuine reader-tag protocol for which the adversary cannot change transmissions (*reader-tag session*). The adversary can access all interfaces concurrently and in many sessions (sessions share a secret key, but have different random tapes). Each session has an identifier sid (given to the adversary, but not to protocol participants). We assume that the adversary knows if an authentication attempt succeeded or not.[4]

In our concurrent single-reader-single-tag scenario (as opposed to a single reader and multiple tags), many instances of the single tag may exist in parallel, sharing the secret key, but not the random tape. The key is *static*, i.e., not updated after executions. For many independent keys (multiple tags), adversaries can always pick a tag to attack in our model. Three factors are crucial to multiple-tag scenarios: the interdependency of the keys; the noise in the communication due to tag-to-reader collisions (a factor modelled by $E_{\max}$); and key management. We cover key management (in particular key updates) in Chapter 3.

We assume message-driven attacks, i.e., honest parties reply as soon as they receive a (protocol) message. The adversary schedules message delivery to honest parties. We assume a global clock, assigning an integer $\mathsf{clock}(\mathsf{sid}, k)$ to the k-th protocol message, delivered in session sid to an honest party. The honest party's reply is assigned $\mathsf{clock}(\mathsf{sid}, k+1) = \mathsf{clock}(\mathsf{sid}, k) + 1$.[5] Furthermore, $\mathsf{clock}(\mathsf{sid}, k) < \mathsf{clock}(\mathsf{sid}^*, k)$ if the adversary delivers the k-th message in session sid^* after the k-th message in session sid. Denote by $\Pi_{\mathsf{sid}}[i \ldots j]$ messages i to j exchanged in session sid and by $\Pi_{\mathsf{sid}}[1 \ldots]$ all messages exchanged in sid. Let $\mathsf{view}_{\mathcal{A}}$ denote the adversary's view in an attack, containing its internal randomness and all the transcripts (of communication with and among other parties).

Let t denote the adversary's running time, including steps of honest parties. Denote by $q_{\mathcal{V}}$ (resp. $q_{\mathcal{P}}$ and q_{OBS}) the maximal number of reader-adversary (resp. adversary-tag and reader-tag) sessions. Below we refine the attacks and define winning conditions for the adversary (who must non-trivially impersonate the tag in a reader-adversary session). For an attack *att* we write $\mathbf{Adv}^{\mathsf{att}}_{\mathsf{DB}}(\mathcal{A})$ for the probability that the $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-adversary $\mathcal{A}$ wins.

Mafia Fraud Resistance. Mafia fraud adversaries can communicate arbitrarily with the tag and reader, *except for purely relaying time-critical transmissions*. We exclude only attacks where the adversary relays *exact* transmissions, calling such time-critical phases *tainted*:

Definition 2.2 (Tainted Time-Critical Phase (Mafia)) *A time-critical phase* $\Pi_{\mathsf{sid}}[k \ldots k+2\ell-1] = (m_k, \ldots, m_{k+2\ell-1})$ *for* $k, \ell \geq 1$ *of a reader-adversary session* sid*, with the* k*-th message being received by the adversary, is* tainted by the

[3]Passive RFID tags have no power source of their own and are very sensitive to their environment, in particular metals and liquids. Semi-passive tags use their own power source for computation, but rely on readers for communication, and are also vulnerable to interference by metals and liquids.

[4]This is not a strong requirement. In practice the success of an authentication attempt is marked by a physical event: a beep, the opening of a door, a green light etc. In the course of this paper, if we assume that an adversary is *not* assumed to know the result of the authentication attempt, this is indicated explicitly —see Section 3.

[5]We could also allow adversaries to delay message delivery *from* honest parties. Our model and results are robust with respect to this idea, but this contradicts the implementation of reliable time measurements and enable denial-of-service attacks.

phase $\Pi_{\mathsf{sid}^*}[k \dots k+2\ell-1] = (m^*_k, \dots, m^*_{k+2\ell-1})$ of an adversary-tag session sid^* *if for all* $i = 0, 1, \dots, \ell-1$ *we have:*

$$\begin{aligned} (m_k, \dots, m_{k+2\ell-1}) &= (m^*_k, \dots, m^*_{k+2\ell-1}), \\ \mathit{clock}(\mathsf{sid}, k+2i) &< \mathit{clock}(\mathsf{sid}^*, k+2i), \\ \textit{and} \qquad \mathit{clock}(\mathsf{sid}, k+2i+1) &> \mathit{clock}(\mathsf{sid}^*, k+2i+1). \end{aligned}$$

Our notion, shown in Figure 8, is conservative with respect to the following:

- We do not exclude phases where the adversary changes the content before relaying. The reason is that since the content may determine the response time if say, the computation is less involved for a 0-bit than for a 1-bit, this may allow the adversary to receive the tag's answer for a different value in time, before it is required to answer to the reader.

  As a consequence, if a protocol sends redundancy like an extra 0-bit, then an adversary can easily flip this bit and not taint the phase, albeit simply relaying the crucial information. We nevertheless grant this freedom to the adversary, as it coincides with the similar idea of matching sessions in key exchange protocols [8, 9]: protocols with obvious redundancy can be easily modified; also, it is common cryptographic practice to err on the safe side.
- According to our definition, a time-critical phase becomes tainted if there is another session in which the adversary relays *all* transmitted messages in the two sessions. If the adversary changes the content of a single transmission in such a phase or the order of a single step only, then the phase is not tainted by the other session anymore. This provides again more freedom to the adversary and strengthens the security notion.

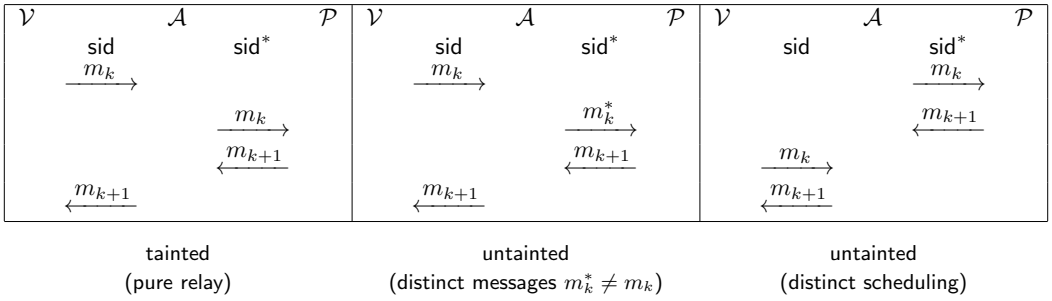

Figure 8: Examples of tainted and untainted time-critical phases

The adversary must now make the reader accept in session sid such that for each adversary-tag session sid^* at most $T_{\max}$ phases of sid are tainted by sid^*:

Definition 2.3 (Mafia Fraud Resistance) *For a distance-bounding authentication scheme* DB *with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$*, a* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-mafia-fraud adversary* $\mathcal{A}$ *wins against* DB *if the reader accepts in one of the* $q_{\mathcal{V}}$ *reader-adversary sessions* sid *such that any adversary-tag session* sid^* *taints at most* $T_{\max}$ *time-critical phases of* sid*. Let* $\mathbf{Adv}^{\mathit{mafia}}_{\mathsf{DB}}(\mathcal{A})$ *denote the probability that* $\mathcal{A}$ *wins.*

Different adversary-tag sessions may taint different rounds of reader-adversary session sid. As we count $T_{\max}$ over all adversary-tag sessions the adversary wins if it taints at most $T_{\max}$ *distinct* phases. Protocols must prevent such attacks to be mafia fraud secure in concurrent settings. Further session interdependencies should also be avoided so that messages from another session do not taint sid.
Note in particular that our definition of mafia fraud resistance allows an adversary to taint more than $T_{\max}$ time-critical sessions of other verifier-adversary sessions than the winning one. In particular, our model *does* allows for mafia fraud key-learning attacks, where the adversary flips a few bits in honest communication sessions (i.e. it taints a large number of time-critical phases, while the prover is in the proximity of the verifier) and thus learns the key bit by bit. Our definition indeed captures such an attack (as opposed to previous models by e.g. Avoine et al. [4]), thus minimizing the restrictions on the adversary's strategies. In practice our definition allows for an adversary placed in the verifier's proximity to forward communication between the prover and verifier (without overstepping the time bound $t_{\max}$) while the prover is also in proximity of the verifier. However, the essence of a mafia fraud attack is that an adversary manages to authenticate from the verifier's proximity, while the prover is *not* in proximity.

Terrorist Fraud Resistance. In a terrorist fraud attack, the tag helps the adversary in all short of revealing its secret key, in fact wanting to ensure that the adversary only wins with the tag's aid (the dishonest prover controls the adversary's access). Desmedt [24] concretely describes the tag's involvement as offline help in a single impersonation attempt. The adversary now wins if the reader accepts, but the adversary cannot use the help given by tag $\mathcal{P}'$ to impersonate further. Note that we denote dishonest provers with an apostrophe, as opposed to our notations in the mafia fraud model.
We formalise the idea by using ideas from proofs of computational ability [7, 76], which exactly capture the intuition of terrorist fraud attacks: given support from a prover, e.g. $\mathcal{P}'$, one can solve a hard problem e.g. identifying to the verifier. This is independent of how the prover gives support. We are not, however, interested in the cases where $\mathcal{P}'$ yields the entire key (or large parts of it) and mark certain auxiliary data given by $\mathcal{P}'$ as trivial, i.e. the data is trivial, if it allows one to successfully complete a "fresh" authentication attempt *without help from* $\mathcal{P}'$. This includes the case when $\mathcal{P}'$ gives the secret key, but circumvents the problem of determining which parts of the key are helpful. Data is trivial if it aids authentication beyond the dedicated help in the session where $\mathcal{P}'$ helps.
We formalise the idea of trivial and non-trivial help by demanding that no algorithm $\mathcal{S}$, called simulator, can use the data passed by $\mathcal{P}'$ to $\mathcal{A}$ to authenticate without the help of $\mathcal{P}'$ (to be fair, we allow $\mathcal{S}$ the same number $q_{\mathcal{V}}$ of attempts as $\mathcal{A}$). This is in line with well-known simulation paradigms, and allows to compare the respective success probabilities of the adversary $\mathcal{A}$ aided by $\mathcal{P}'$, and the simulator $\mathcal{S}$ using $\mathcal{A}$'s information to authenticate. If $\mathcal{A}$ is significantly more successful than $\mathcal{S}$, the attack is non-trivial and the protocol is insecure against terrorist attacks. Note that "unsophisticated" adversaries may do worse than simulators for secure schemes, thus yielding negative advantages.
For terrorist fraud, $\mathcal{A}$ acts as for mafia fraud, but may query the "malicious" interface $\mathcal{P}'$ in lazy phases. Sessions sid′ with $\mathcal{P}'$ are arbitrary, not following protocol. In fact we may consider only one session sid′ when $\mathcal{P}'$ helps $\mathcal{A}$. The tag may *not* aid $\mathcal{A}$ in time-critical phases, a fact which we model by defining tainted time-critical phases as pure-relay phases or rounds where $\mathcal{A}$ queries $\mathcal{P}'$.

Definition 2.4 (Tainted Time-Critical Phase (Terror)) *A time-critical phase* $\Pi_{\mathsf{sid}}[k \dots k+2\ell-1] = (m_k, \dots, m_{k+2\ell-1})$ *for* $k, \ell \geq 1$ *of a reader-adversary session* sid*, with the* k*-th message being received by the adversary, is* tainted *if there is a session* sid′ *between the adversary and* $\mathcal{P}'$ *such that, for some* i*,*

$$\mathit{clock}(\mathsf{sid}, k) < \mathit{clock}(\mathsf{sid}', i) < \mathit{clock}(\mathsf{sid}, k + 2\ell - 1).$$

For the new definition of tainted phases, terrorist fraud resistance demands that for any terrorist fraud attacker $\mathcal{A}$ there exists a simulator $\mathcal{S}$ such that for any supporting $\mathcal{P}'$, $\mathcal{S}$ is essentially as successful as $\mathcal{A}$. We use concrete security statements and omit quantification over $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{P}'$ algorithms; this quantification is included in subsequent security claims in the usual form (i.e. for any adversary there exists a simulator such that for all tags the advantage is small).
As explained above, the simulator gets as input only the adversary's randomness and the transcript of all communication exchanged during a run of $\mathcal{A}$'s attack and must pass one of at most $q_{\mathcal{V}}$ authentication attempts given only this data. For example, if $\mathcal{P}'$ hands over the secret key, then $\mathcal{S}$ has a trivial task. However, for more advanced strategies, the simulator may need to work harder. A scheme is terrorist fraud resistant if for any adversary which succeeds, with some probability, to authenticate with the prover's help, there exists a simulator that, given the adversary's internal view, authenticates with the same probability. We can also think of attacks for which such a simulator exists as invalid.
Note that in this definition *any* non-negligible loss of entropy regarding the secret key through the auxiliary data invalidates the terrorist attack. To see this, consider the example of an adversary that is given a (significant) number k of bits of the secret key (which has a size of n bits in total). The adversary may then guess the rest of the key with probability $\frac{1}{2^{n-k}}$ and pass authentication in a single attempt. However, afterwards the simulator uses the adversary's view and succeeds with the same probability as the adversary, also by guessing. In our definition, the existence of such a simulator makes this specific terrorist fraud attack invalid in the sense that it would count as trivial.

Definition 2.5 (Terrorist Fraud Resistance) *Let* DB *be a distance-bounding authentication scheme with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$*. Let* $\mathcal{A}$ *be a* $(t, q_{\mathcal{V}}, q'_{\mathcal{P}})$*-terrorist-fraud adversary,* $\mathcal{S}$ *be an algorithm running in time* $t_{\mathcal{S}}$*, and* $\mathcal{P}'$ *be an algorithm running in time* t'*. Denote*

$$\mathbf{Adv}^{\mathit{terror}}_{\mathsf{DB}}(\mathcal{A}, \mathcal{S}, \mathcal{P}') = p_{\mathcal{A}} - p_{\mathcal{S}}$$

where $p_{\mathcal{A}}$ *is the probability that the reader accepts in one of the* $q_{\mathcal{V}}$ *reader-adversary sessions* sid *such that at most* $T_{\max}$ *time-critical phases of* sid *are tainted, and* $p_{\mathcal{S}}$ *is the probability that, given* $\mathit{view}_{\mathcal{A}}$ *in an attack of* $\mathcal{A}$*,* $\mathcal{S}$ *makes the reader accept in one of* $q_{\mathcal{V}}$ *subsequent executions.*

Again, if the advantage is negative, the adversary $\mathcal{A}$ performs worse than $\mathcal{S}$. Our notion is quite strong: the simulator only gets to see $\mathcal{A}$'s transcript in an offline phase, instead of communicating with $\mathcal{P}'$ online. This guarantees stronger security and saves us from dealing with issues related to the number of queries and successful attacks (adversary vs. simulator).
How does our definition fit into previous efforts? Previous protocols [6, 70] claim to achieve a security of $(1/2)^{-N_c}$. This, however, corresponds to a tailor-made strategy of $\mathcal{P}'$; other strategies may still exist. Proving that the advantage in Definition 2.5 is negligible, then we *prove* that $\mathcal{P}'$ can only help trivially.

Distance Fraud Model. For distance fraud, an adversary must reply ahead of a time-critical phase or it cannot respond in time. In practice this is enforced by a tight value of $t_{\max}$. For any time-critical phase, with possibly many communication rounds, the adversary must commit to the *first* message to be sent. For any later rounds in the phase, the adversary has time to reply even from farther away.
The order of committed and sent values is determined by on oracle CommitTo with a single session $\mathsf{sid}_{\mathsf{CommitTo}}$, taking tuples (sid, i, m_i) from the adversary and giving empty responses. The adversary commits to the first message of time-critical phase i of session sid (message j in sid) at time $\mathsf{clock}(\mathsf{sid}_{\mathsf{CommitTo}}, j)$. As the adversary may repeatedly commit to this message, we take the last commitment before phase i begins. A time-critical phase is tainted if the adversary returns an answer it has not committed to.

Definition 2.6 (Tainted Time-Critical Phase (Distance)) *A time-critical phase* $\Pi_{\mathsf{sid}}[k \dots k{+}2\ell{-}1] = (m_k, \dots, m_{k+2\ell-1})$ *for* $k, \ell \geq 1$ *of a reader-adversary session* sid*, with the* k*-th message being received by the adversary, is* tainted *if the maximal* j *with* $\Pi_{\mathsf{sid}_{\mathit{CommitTo}}}[j] = (\mathsf{sid}, k+1, m^*_{k+1})$ *for some* m^*_{k+1} *and clock*$(\mathsf{sid}, k) >$ *clock*$(\mathsf{sid}_{\mathit{CommitTo}}, j)$ *satisfies* $m^*_{k+1} \neq m_{k+1}$ *(or no such* j *exists).*

Definition 2.7 (Distance Fraud Resistance) *For an authentication scheme* DB *with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$*, a* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-distance-fraud adversary* $\mathcal{A}$ *wins against* DB *if the reader accepts in one of* $q_{\mathcal{V}}$ *reader-adversary sessions* sid *with at most* $T_{\max}$ *tainted time-critical phases.Let* $\mathbf{Adv}^{\mathit{dist}}_{\mathsf{DB}}(\mathcal{A})$ *be the probability of* $\mathcal{A}$ *winning.*

Impersonation Resistance. We suggest a simple, but very strong definition of impersonation security as a basic requirement of authentication in our concurrent setting. Thus even adversaries who actively take part in concurrently-run prover and verifier sessions cannot impersonate the prover. Whereas the previous properties concern time-critical phases, impersonation security requires that an adversary cannot impersonate a tag in *lazy* phases. This ensures that the reader leaks no time-critical information to an invalid tag. Following the idea that parties should authenticate even if the time-critical phases are not executed, we consider projections $\Pi^{\mathsf{lazy}}_{\mathsf{sid}}[1 \dots]$ of $\Pi_{\mathsf{sid}}[1 \dots]$ containing lazy phases transmissions only, and (not necessarily consecutive) indices $\iota^{\mathsf{lazy}}_{\mathsf{sid}} = (i_1, i_2, \dots)$ of lazy-phase messages. The adversary wins if a reader-adversary session succeeds and no adversary-tag session has the same "lazy transcript", created via pure relaying.

Definition 2.8 (Impersonation Security) *In a distance-bounding authentication scheme* DB *with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$ *where* $\mathcal{V}$ *always goes first, a* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-impersonation adversary* $\mathcal{A}$ *wins against* DB *if* $\mathcal{V}$ *accepts in a reader-adversary session* sid *such that no adversary-tag session* sid^* *has*

$$\Pi^{\mathit{lazy}}_{\mathsf{sid}}[1 \dots] = \Pi^{\mathit{lazy}}_{\mathsf{sid}^*}[1 \dots],$$

and

$$\mathit{clock}(\mathsf{sid}, i) < \mathit{clock}(\mathsf{sid}^*, i)$$

for any $i \in \iota^{\mathit{lazy}}_{\mathsf{sid}} \cap \iota^{\mathit{lazy}}_{\mathsf{sid}^*}$ *s.t.* $\mathcal{V}$ *has sent the* i*-th message to* $\mathcal{A}$ *in* sid*, and*

$$\mathit{clock}(\mathsf{sid}, j) > \mathit{clock}(\mathsf{sid}^*, j)$$

for any $j \in \iota^{\mathit{lazy}}_{\mathsf{sid}} \cap \iota^{\mathit{lazy}}_{\mathsf{sid}^*}$ *such that the adversary has sent the* j*-th message to the reader in* sid*. Let* $\mathbf{Adv}^{\mathit{imp}}_{\mathsf{DB}}(\mathcal{A})$ *be the probability that* $\mathcal{A}$ *wins.*

2.1.3 Relating the Models

Impersonation security concerns lazy protocol phases, while terrorist, mafia, and distance fraud attack time-critical phases. In our framework we refute the idea in [70] that terrorist fraud resistance implies distance fraud resistance and show that all properties are independent. The formal proofs for each statement are shown below, after we briefly discussed the freshly-introduced security definitions.

Theorem 2.9 (Security Diagram — Informal) *If pseudorandom functions exist, the following holds:*

1. *There exists a distance-bounding authentication scheme that is impersonation-secure, mafia and distance fraud resistant, but* not *terrorist fraud resistant.*
2. *There exists a distance-bounding authentication scheme that is impersonation-secure, terrorist and mafia fraud resistant, but* not *distance fraud resistant. Thus, terrorist fraud resistance does not imply distance fraud resistance.*
3. *There exists a distance-bounding authentication scheme that is impersonation-secure, terrorist and distance fraud resistant, but* not *mafia fraud resistant. Thus, terrorist fraud resistance does not imply mafia fraud resistance.*

Terrorist Fraud Resistance. The enhanced Kim-Avoine scheme in Section 2.3.4 has all properties except for terrorist fraud resistance. The reason it fails against terrorist attacks is that time-critical messages are predetermined by the lazy phase and can be revealed without disclosing the secret key (thus providing sufficient, but non-trivial offline help). In general, terrorist attacks are thwarted by interlinking authentication sessions, such that malicious tags (partially) reveal long-term secrets if they help the adversary. The difficulty in designing terrorist fraud resistant schemes is formally ensuring that the simulator can extract the secret from the adversary and thus authenticate. The simulator's only advantage is that it can rewind executions and get responses for different challenges.

Distance Fraud Resistance. We separate distance fraud resistance from the other properties by giving the tag a special key which makes time-critical responses predictable. Honest parties never use this key, but malicious tags may use it to commit distance fraud. Other security properties are unaffected, as the special key is never used by honest parties. Distance fraud resistance depends on the unpredictability of each round's answer. This is easily achieved by adding some time-critical rounds where tags echo random bits. Note that this separation begins from the assumption that there exists a distance-fraud resistant distance-bounding protocol. In view of the recent work of Boureanu et al. [11], it is not clear that such a protocol exists.

Mafia Fraud Resistance. We show the independence of mafia fraud resistance from the other security notions by starting with a protocol having all other security properties; the tag may use a bit to indicate that time-critical bits are flipped. Then a man-in-the-middle (MITM) adversary can flip replies from an adversary-tag session and authenticate to the reader without tainting the phases. There are two options to prevent mafia fraud attacks. Assume that in each fast phase the reader sends a random challenge. If the adversary correctly predicts the challenge in a reader impersonation, it can use the reply in the reader-adversary session without tainting the phase; for a wrong prediction, the adversary guesses the answer instead. The overall success is $\frac{3}{4}$ per round as in, e.g. the Hancke and Kuhn protocol [42]. The other option is to authenticate the reader by the fast phase challenges. Now the adversary-tag session in the above attack aborts for a wrong prediction, dropping the adversary's success probability in the reader-adversary execution to $\frac{1}{2}$ for subsequent rounds. This is the strategy of the Kim-Avoine protocol [49].
In what follows, we formally state and prove the relationships between our security notions.

Proposition 2.10 *If a pseudorandom function $\mathcal{PRF}$=(Kg, PRF) exists, then there exists a distance-bounding authentication scheme $\mathcal{ID}_{terror}^{imp,dist,mafia} = (\mathsf{Kg}_{\mathsf{PRF}}, \mathcal{V}_{\mathsf{PRF}}, \mathcal{P}_{\mathsf{PRF}})$ with parameters $(t_{\max}, T_{\max}, E_{\max}, N_c)$ that is mafia and distance fraud resistant, and secure against impersonations, but that is vulnerable to terrorist fraud attacks.*

Proof. This scheme is the enhanced Kim/Avoine scheme shown in Section 2.3.4. □

Proposition 2.11 *If a distance-bounding authentication scheme $\mathcal{ID}'$ exists such that it is resistant to mafia and terrorist fraud, and impersonation resistant, then there exists a distance-bounding authentication scheme $\mathcal{ID} = \mathcal{ID}_{dist}^{imp,mafia,terror} = (\mathsf{Kg}, \mathcal{V}, \mathcal{P})$ with parameters $(t_{\max}, T_{\max}, E_{\max}, N_c)$ that is still secure against impersonations and terrorist and mafia fraud resistant, but that is vulnerable to distance fraud adversaries. In particular, terrorist fraud resistance does not imply distance fraud resistance.*

Proof. Consider an authentication scheme $\mathcal{ID}'$ that is mafia and terrorist fraud resistant and secure against impersonations. Modify $\mathcal{ID}'$ to obtain $\mathcal{ID}$ as follows: apart from any secret(s) the tag and the reader share in $\mathcal{ID}'$, add another secret key sk^*. In the lazy phase, the tag's first message to the reader will now be preceded by a bit b and a bitstring V of length $|sk^*|$. An honest tag always sends $b = 0$ and $V = \mathbf{0}$, the all-zero vector. The reader parses the beginning bit and checks it. If the received bit is 0, protocol $\mathcal{ID}'$ is followed exactly, in its original form (and V is ignored). Else, if the received bit is 1, the reader skips any checks on the tag's lazy phase messages that appear in $\mathcal{ID}$ and checks that $V = sk^*$. If so, the reader goes on to the time-critical phases and always expects a 0 response from the tag if the response comes in time.
The following statements hold for DB:

- For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-impersonation adversary $\mathcal{A}$ against DB there exists a $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-impersonation adversary $\mathcal{A}'$ against $\mathcal{ID}'$ such that

$$\mathbf{Adv}^{\mathsf{imp}}_{\mathsf{DB}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{imp}}_{\mathcal{ID}'}(\mathcal{A}') + q_{\mathcal{V}} \cdot 2^{-|sk^*|}.$$

- For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-mafia adversary $\mathcal{A}$ against DB there exists a $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-mafia adversary $\mathcal{A}'$ against $\mathcal{ID}'$ such that

$$\mathbf{Adv}^{\mathsf{mafia}}_{\mathsf{DB}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{mafia}}_{\mathcal{ID}'}(\mathcal{A}') + q_{\mathcal{V}} \cdot 2^{-|sk^*|}.$$

- For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q'_{\mathcal{P}}, q_{\text{OBS}})$-terrorist-fraud adversary $\mathcal{A}$ against DB there exists a $t_{\mathcal{S}}$-simulator $\mathcal{S}$ such that for any $\mathcal{P}'$ running in time $t_{\mathcal{P}'}$ it holds that,

$$\mathbf{Adv}^{\mathsf{terror}}_{\mathsf{DB}}(\mathcal{A}, \mathcal{S}, \mathcal{P}) \leq 0.$$

- This scheme is not resistant to distance fraud.

The first bound follows from the bound of the underlying construction. Honest users do not send sk^*, therefore an adversary against impersonation does not know sk^*. An impersonation adversary $\mathcal{A}^{\mathsf{imp}}_{\mathsf{DB}}$ against the distance-bounding authentication scheme DB must either break the underlying schema $\mathcal{ID}'$ or guess the secret key sk^*. This adds a term $2^{-|sk^*|}$ to the impersonation security bound for each authentication attempt.
Similarly, honest participants will never send 1 in the first message and subsequently $V = sk^*$. Therefore, a mafia fraud adversary $\mathcal{A}^{\mathsf{mafia}}_{\mathsf{DB}}$ against DB can either guess sk^* (and succeed with probability 1) or can try to break the underlying construction. Therefore, the advantage of a mafia adversary against DB is the advantage of a mafia adversary against $\mathcal{ID}'$ with an added term $2^{-|sk^*|}$.
Let $\mathcal{A}^{\mathsf{terror}}_{\mathsf{DB}}$ be an adversary against the terrorist fraud resistance of scheme DB. We build a simulator $\mathcal{S}_{\mathsf{DB}}$ as follows. The simulator looks first in the transcripts of $\mathcal{A}^{\mathsf{terror}}_{\mathsf{DB}}$; if sk^* was used, the simulator reuses this value and wins with probability 1. If sk^* was not used, the simulator removes the first bit from each execution, and also the string V, then following the protocol of $\mathcal{ID}'$. The adversary $\mathcal{A}^{\mathsf{terror}}_{\mathsf{DB}}$ is now an adversary against $\mathcal{ID}'$ and therefore there must exist a simulator $\mathcal{S}^*$ such that $p_{\mathcal{A}^{\mathsf{terror}}_{\mathsf{DB}}} - p_{\mathcal{S}^*} \leq 0$. The simulator $\mathcal{S}$ against DB uses $\mathcal{S}^*$ against $\mathcal{ID}'$ as a black box, and adds a 0 bit and the string $V = \mathbf{0}$ to the execution. Clearly, $p_{\mathcal{S}} = p_{\mathcal{S}^*}$. Therefore the bound in this statement is achieved.
An adversary $\mathcal{A}^{\mathsf{dist}}_{\mathsf{DB}}$ against DB is a legitimate tag, which therefore knows the value of sk^*. This adversary will send as its first message 1 and later $V = sk^*$. During the fast phase, the adversary will commit to each round a response of 0. This adversary succeeds with probability 1.

□

Proposition 2.12 *If a distance-bounding authentication scheme $\mathcal{ID}'$ exists such that it is secure against impersonations and resistant to terrorist and distance fraud, then there exists a distance-bounding authentication scheme* $\mathsf{DB} = \mathcal{ID}^{imp,dist,terror}_{mafia} = (\mathsf{Kg}, \mathcal{V}, \mathcal{P})$ *with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$ *that is still secure against impersonations and resistant to terrorist and distance fraud, but that is vulnerable to mafia fraud adversaries. In particular, terrorist fraud resistance does not imply mafia fraud resistance.*

Proof. Consider an authentication scheme $\mathcal{ID}'$ that is secure against impersonations and resistant to terrorist and distance fraud. Modify $\mathcal{ID}'$ to obtain $\mathcal{ID}$ as follows: apart from any secret(s) the tag and the reader share in $\mathcal{ID}'$, add another secret key sk^*. In the lazy phase, the tag's first message to the reader will now be preceded by a bit b. An honest tag always sends $b = 0$. The reader parses the beginning bit and checks it. If the received bit is 0, protocol $\mathcal{ID}'$ is followed exactly, in its original form. Assume that in the N_c time-critical rounds in $\mathcal{ID}'$, the tag sends responses T_i, which are verified by the reader. If the bit received by the reader during the lazy phase of $\mathcal{ID}$ is $b = 1$, in each of the time-critical rounds of this protocol, the reader will expect response $\bar{T}_i$, i.e. the bit(s) of the response are flipped.
The following statements hold for DB:

- For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-impersonation adversary $\mathcal{A}$ against DB there exists a $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-impersonation adversary $\mathcal{A}'$ against $\mathcal{ID}'$ such that

$$\mathbf{Adv}^{\mathsf{imp}}_{\mathsf{DB}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{imp}}_{\mathcal{ID}'}(\mathcal{A}').$$

- For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-distance adversary $\mathcal{A}$ against DB there exists a $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-distance adversary $\mathcal{A}'$ against $\mathcal{ID}'$ such that

$$\mathbf{Adv}^{\mathsf{dist}}_{\mathsf{DB}}(\mathcal{A}) \leq \mathbf{Adv}^{\mathsf{dist}}_{\mathcal{ID}'}(\mathcal{A}').$$

- For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q'_{\mathcal{P}}, q_{\text{OBS}})$-terrorist-fraud adversary $\mathcal{A}$ against DB there exists a $t_{\mathcal{S}}$-simulator $\mathcal{S}$ such that for any $\mathcal{P}'$ running in time $t_{\mathcal{P}'}$ it holds that,

$$\mathbf{Adv}^{\mathsf{terror}}_{\mathsf{DB}}(\mathcal{A}, \mathcal{S}, \mathcal{P}) \leq 0.$$

- This scheme is not resistant to mafia fraud.

The first bound follows from the bound of the underlying construction, as the same adversary that succeeds in an impersonation attack against $\mathcal{ID}'$ will succeed in an impersonation attack against $\mathcal{ID}$.
We consider the second statement. Let $\mathcal{A}^{\mathsf{dist}}_{\mathsf{DB}}$ be a distance-fraud adversary that succeeds against the authentication scheme $\mathcal{ID}$. We show how to use this adversary to construct a distance-fraud adversary $\mathcal{A}^{\mathsf{dist}}_{\mathcal{ID}'}$ against the underlying scheme $\mathcal{ID}'$. If $\mathcal{A}^{\mathsf{dist}}_{\mathsf{DB}}$ forwards a 0 prepended to its first lazy-phase message, the adversary $\mathcal{A}^{\mathsf{dist}}_{\mathcal{ID}'}$ forwards the exact responses T_i which $\mathcal{A}^{\mathsf{dist}}_{\mathcal{ID}}$ commits to in each round. Otherwise, if a 1 is used during the lazy-phase, the adversary simply flips the bits.
Let $\mathcal{A}^{\mathsf{terror}}_{\mathsf{DB}}$ be an adversary against scheme DB. We build a simulator $\mathcal{S}$ as follows. The simulator removes the $0/1$ bit from each execution, following the protocol of $\mathcal{ID}'$. The adversary $\mathcal{A}^{\mathsf{terror}}_{\mathsf{DB}}$ is now an adversary against $\mathcal{ID}'$ and therefore there must exist a simulator $\mathcal{S}^*$ such that $p_{\mathcal{A}^{\mathsf{terror}}_{\mathsf{DB}}} - p_{\mathcal{S}^*} \leq 0$. The simulator $\mathcal{S}$ against DB uses $\mathcal{S}^*$ against $\mathcal{ID}'$ as a black box, and adds a 0 bit to the execution. Clearly, $p_{\mathcal{S}} = p_{\mathcal{S}^*}$. Therefore the bound in this statement is achieved.
An adversary $\mathcal{A}^{\mathsf{mafia}}_{\mathsf{DB}}$ against DB opens a reader-adversary session sid and an adversary-tag session sid*. During the lazy phase, the adversary relays messages from one session to another (this is permitted during the lazy phase), but in the first message flips the 0 bit sent by the honest tag in sid* to a 1 bit. During each of the time-critical rounds, the adversary relays the challenge bits of the reader, but flips the return bits T_i sent by the tag. Since the communication is not simply relayed, this is a valid mafia adversary, which wins with probability 1. □

2.1.4 RFID Distance-Bounding in Practice

So far we have discussed both theoretical (modelling) aspects of distance-bounding and more practical mafia and terrorist fraud resistant constructions. In this paragraph we elaborate on the actual implementation of such constructions. Towards this goal, we first summarise distance-bounding properties of RFID tags, and then give general pointers regarding cryptography on RFID tags.

Tag Flavours: HF, UHF, and Microwaves. The most common standard for RF tags (or labels) is the ISO 14443 standard, which describes two types of tags operating at a a frequency of 13.56 MHz. This standard is implemented e.g. in MIFARE and LEGIC cards. The ISO 14443 standard describes so-called HF (high frequency) cards, characterised by medium data transfer rates and by sensitivity to metals and liquids [21]. These cards operate at proximity, i.e., at around 10 cm.
Hancke and Kuhn [42] argued that distance-bounding protocols are unsuitable for HF tags, instead suggesting an implementation on so-called Ultra-Wide Band (UWB) devices, i.e., devices with a frequency of over 500 MHz. For RFID, this class contains UHF and Microwave devices, characterised by high and very high transfer rates, as well as an even higher vulnerability to metals and liquids. Both [42] and [20] show how adversaries may benefit from time slots allowed for transmission of redundant tag data, or how they may exploit timing tolerance in the coding and modulation stages. It is concluded in [41] that distance-bounding protocols are only implementable in systems with low-latency channels.
Reid et al. [70] point out that distance bounding protocols require unconventional RFID system properties, namely very low communication latency, but not necessarily a high bit rate (as only bits are exchanged during time-critical phases). The goal is to isolate propagation delays from measured times consisting of twice the propagation delay and the processing delay for the tag. Propagation delays may only be isolated if processing delays are invariant and negligibly low. As ISO 14443 typically operates on a proximity of 10 cm, the propagation time is very small and round-trip time-of-flight take about $2/3$ ns. However, the clock rate of HF readers has cycles of 74 ns, enough for the signal to propagate over 22 m. If the processing time also varies, it can be used by adversaries to mask relaying [70]. A UWB implementation measures time more accurately due to high bit rates; however, UWB implementation is expensive [70] and it is thus worthwhile to investigate cheaper alternatives.
Indeed, [70] argue that the most significant delay in a relay attack is due to processing delays in the relay devices; such relays can be detected even on HF tags if the subcarrier communication in the time-critical phases is replaced by simple peak detection on the main carrier [70]. However, both [70] and [41] note that this approach assumes that tags are protected against overclocking attacks, where the adversary attempts to speed up the tag's processing time (thus learning the response earlier from the tag and tainting each round) by increasing the externally-supplied clock. Another assumption is that the RF carrier operates within a small tolerance margin specified by the relevant standards. As [41] notes, however, neither condition applies to the current tag standards.

Cryptography on symmetric-key RFID Tags. Apart from passive, semi-passive, and active, tags can also be classified in terms of their functionality (and thus also by the type of cryptographic primitives they support). This classification was proposed by the former Auto-ID Center and includes [21] four classes:

Class I. These passive tags have minimal required functionality and are equipped with a unique, read-only identifier. The tag sends this identifier in clear every time it is prompted by a reader. Additionally, the tag contains a kill password (which readers use to permanently disable the tag) and a CRC for transmission verification. Such tags are generally used in logistics and product tracking. The most commonly used Class I tags are EPC Class I Generation 2 tags.

Class II. These tags are slightly more expensive than Class I tags, and are equipped with read-write capabilities. Their functionality features also data security, privacy, and theft detection, and though they are passive and usually used only for proximity, these tags are also able to perform slightly more complex operations.

Class III. These tags are semi-passive or active, containing more advanced and dedicated (also, more expensive) hardware and battery support for long-range communication. Such tags may also support broadband communication, such as UWB.

Class IV. Finally, these tags are able to network in peer-to-peer networks with other Class IV tags when the same frequency is used. They are the most complex and expensive type of tags, are active, and they are able to form ad-hoc networks.

Arguably, RFID technology has gained so much support in recent years due to its small size and low price, which allows the technology to be widely employed in, for instance, logistics. It is also paramount, however, to achieve at least a minimal level of security and privacy in RF systems. EPC Class I Generation 2 tags employ UHF technology, and they are the most widely used RF tags in existence. Due to their simplicity and to the fact that such tags always respond to a reader prompt by simply giving their unique EPC identifier in clear (or they self-destruct if they receive a kill password), EPC Class I Generation 2 tags have small surface area and are practically negligibly expensive. However, they do not preserve almost any security and privacy requirements.
The ability to implement some kind of private-key primitive —in practice, the lightweight HMAC algorithm is an obvious option and believed to realise a pseudorandom function (PRF)— is a minimum requirement of private authentication. Current passive tag designs are incompatible with such implementations, which require about 8120 GE (gate equivalents) [10, 29] for SHA-1 and even more GE for SHA-256. The larger the number of GE, the larger the chip area must be, and therefore the higher the production costs. However, increased interest in RFID authentication has resulted in fast improvements in RFID tag design technology — we mention here the recent work by Lee et al. [52], Leinweber et al. [53], and Wenger and Hutter [75], which show processor designs for RFID tags which can more efficiently compute cryptography on elliptic curves. Thus, future processor designs may enable either better HMAC implementations (by using dedicated hardware) or different, public-key primitives enabling distance bounding. Additionally, manufacturing unit costs for RFID tags have dropped dramatically in recent years, from around one dollar in 2000 to about 10 cents or less nowadays; we may therefore assume that the more expensive RFID tags of today may become much cheaper in the future.
Such considerations motivate us to use primitives which offer superior privacy properties, such as HMAC, for RFID distance bounding and authentication protocols.

2.2 Related Work

In this section, we first briefly review literature adjacent to our approach towards defining the security of distance-bounding protocols. In particular, we first briefly review three related topics to distance bounding, namely: position-based cryptography, self-delegatable schemes, and the delegation of anonymous credentials; then we explain the relation of our framework with the parallel, less formal framework of Avoine et al. [4]. In the first part, i.e. section 2.2.1 we show exactly how each of the three related topics resemble our models and how they relate to the intuition behind distance-bounding attacks. In the second part, i.e. section 2.2.2 we show that the preceding framework by Avoine et al. considers a different notion of terrorist fraud resistance (this notion is not formally defined, therefore the comparison of the two notions is not very precise). We also show how the notions of mafia, terrorist, and distance fraud resistance are achieved in the literature.

2.2.1 Related Topics to Distance-Bounding

Position-Based Cryptography. Chandran et al. [18] recently introduced the notion of position-based cryptography, where a set of verifiers wishes to check whether a prover is at a position P or not. Apart from broadcasting and sending directional messages to the prover, verifiers can also securely communicate with one other. Provers can only broadcast or send directional messages to a verifier. The so-called vanilla model considered by Chandran et al. [18] involves

several colluding adversaries, which may broadcast, send directional messages, and communicate securely with each other. Communication time is measured according to distance.
Chandran et al. [18] point out that collusion attacks are an important aspect when considering position-based cryptography. They prove an important impossibility result in this setting, namely that secure positioning cannot be achieved in the vanilla model (with collusions). Basically in the proof, colluding adversaries are placed in a circle, centred around point P, where the prover should be closer to the verifiers than P. The adversaries can communicate with one another within time α.Then each adversary impersonates a copy of the prover for the verifier in proximity and answers each message of this verifier on the prover's behalf, but faster since it is closer. The time gained is used to "synchronise" with the other adversaries such that each prover copy is always up-to-date, with a time delay of α. Overall, this collusion strategy successfully simulates the presence of a prover at P.
How does this model (and impossibility result) relate to RFID distance-bounding attacks? The main difference to mafia fraud is that the mafia adversary in our model doesn't know the symmetric key shared between prover and verifier. Even colluding adversaries are thus unable to answer on the tag's behalf quickly and then "synchronise" (the impossibility result seems to require that the adversaries have all the information the prover has). Also, mafia attacks resemble more malleability attacks, as the adversary has access to other copies of the protocol (but with switched roles) and tries to take advantage of this, with some restrictions due to the distance. Position-based cryptography considers instead a single protocol run with multiple verifiers.
If the colluding adversaries in [18] hold the secret key in a setting resembling our distance fraud attacks, each prover copy is in fact closer to the verifiers than point P. The success of such an adversary is in tune with our assumption that legitimate provers may authenticate within close proximity.
It is unclear how position-based cryptography compares to terrorist attacks, where the adversary has some limited help. This idea is closer related to the concept of non-transferability (see below), discouraging users to reveal parts of their secrets [28, 36, 12, 16]. Also, establishing exact position is impossible in practice for RFID tags, as their response times have a high variance. Only by using an extremely large number of readers is this possible.

From Useful Help to Undesired Disclosure. Several (public-key based) approaches in the literature [12, 16, 28, 36] associate the distribution of parts of a secret to leaking (external) personal information like a credit card number [12, 28], or the entire (internal) secret itself [16, 36]. We discuss the latter case, which resembles our terrorist attacks.
Goldreich et al. [36] introduce the idea of self-delegatable schemes, where users generate secondary keys with restricted rights, usable in more vulnerable environments, e.g. laptops; the secondary keys are authenticated through a long-term key. Losing a few such keys should not harm the security of other keys; however, since only self-delegation is supported, leaking too many secondary keys endangers the security of the user's long-term key.
In a sense, the idea behind self-delegatable schemes is mirrored in our terrorist attack resistance: if the support the adversary receives from the tag allows authentication, then this leaks essential information about the tag's secret. The main differences in the model are: that [36] consider the public-key setting only (where server certification of secondary keys is used); that they investigate signature-leakage only[6]; and that no online help (with restriction due to the distance) is available. Also, all the schemes in [36] rely on public-key cryptography and non-interactive zero-knowledge proofs, and are unsuitable for RFID.
Camenisch and Lysanskaya [16] model transferability of anonymous credentials. This "all-or-nothing" approach associates sharing secret information (pseudonyms or credentials only) to recovery of users' full secret. This is again similar to terrorist resistance, but [16] do not formally model attacks and security. The use of public-key infrastructures here also makes the idea inapplicable to RFID.

2.2.2 A Previous Distance-Bounding Framework

Both our framework and that in [50] indicate that there are four main security threats to distance-bounding protocols: mafia fraud, terrorist fraud, distance fraud, and offline impersonation security. We show here the main differences between the two frameworks, and what common means are used to achieve each of the notions in the literature (this brief overview will be found in more detail in the comparison of prominent protocols in Section 2.3).

Mafia Fraud. In [4], mafia fraud is defined as an attack whereby a man-in-the-middle (MITM) adversary "defeats" – i.e. breaks the soundness of – a distance-bounding protocol. We note, however, that this definition, while more formal than the intuitive notions found in the literature before 2009, does not describe in detail which MITM attacks are allowed, nor how they are modelled. In particular, the power of the clock in the distance-bounding protocol is not specified.

[6] They do briefly mention proofs of computational ability as a possibility to implement for fine-grained leakage, though.

By contrast, our approach is to restrict the power of the clock such that it only detects *pure relay only*, as described in Section 2.1. Relaying is defined *only for time-critical rounds*, and in terms of sessions (reader-tag, reader-adversary, adversary-tag). Our notion of tainted sessions describes exactly and formally the limits of the adversary's relaying ability (in particular, we allow the adversary to flip bits and then relay the modified message between honest parties). This latter strategy is not listed by [4]; it is, however, feasible, as a protocol with e.g. fuzzy response verification may allow an adversary to flip bits and thus win against the verifier. In our framework, as presented in Section 2.1, we only minimally restrict the adversary.

Furthermore, we account for so-called key-learning attacks, i.e. attacks for which an adversary relays protocol messages between provers and verifiers (when the prover is in proximity of the verifier), but flips bits in only some time-critical phases, with the express goal of learning a long-term secret. This attack is not covered by the framework of Avoine et al.; however, many distance-bounding schemes aiming to achieve terrorist fraud resistance are susceptible to it. We describe this attack in more detail in Section 2.3.5 and Chapter 4. In particular, we prove that the distance-bounding scheme due to Bussard and Bagga [**?**] is *not* mafia fraud resistant, while the scheme due to Reid et al. [70] cannot be *generically* proved mafia fraud resistance.

The security model of Avoine et al. [4] defines mafia fraud security in terms of soundness, i.e. a mafia adversary succeeds if it breaks the soundness of the protocol in a MITM attack. By contrast we exactly quantify the winning probability, or *advantage*, of the adversary. This allows for an easier comparison of the security of various protocols. We note that the protocols due to [13] and [49] only allow the adversary to succeed in mafia fraud with a probability of approximately $\frac{1}{2}$ per time-critical phase. The protocol due to [42], however, allows an adversary to win with a probability of approximately $\frac{3}{4}$ per time-critical phase. Whereas all these constructions are sound in the model of Avoine et al. [4] for sufficiently many time-critical phases, the protocol due to [42] requires a lot more rounds than the former two protocols in order to reach the same security level. In other words, it is less efficient and therefore less suitable for resource-constrained devices. We refer the reader to Section 2.3 for more details about the exact security properties of several known constructions in the literature, including the scheme due to Hancke and Kuhn [42].

We briefly examine the effects of common strategies, in particular mutual authentication, in achieving mafia fraud resistance. A better protocol analysis follows in Section 2.3. If the time-critical rounds are independent and there is no verifier authentication (as is the case for the protocol in [42]), the adversary will succeed in guessing about a half of the challenges. Furthermore, even for challenges that he does not guess, the adversary still has a probability of about $\frac{1}{2}$ to guess the correct response. However, we show in Section 2.3.4 an improved version of the protocol due to Kim and Avoine [49] which has a much better mafia fraud resistance. As in the original construction of Kim and Avoine, there is some mutual authentication required in the time-critical rounds, i.e., the verifier must also authenticate during these rounds. If the verifier's challenge is not verified by the prover, then the prover responds randomly in that round and in all subsequent rounds. In other words, an adversary as described above can only query the prover *as long as his guesses are correct*. As soon as one guess is incorrect, all subsequent responses have only a $\frac{1}{2}$ probability to verify. The same effect is achieved by the protocol due to [6], but at higher storage costs. Another approach is taken by constructions such as the Swiss-Knife protocol [50], where the challenges are subsequently authenticated in a second lazy phase. Whereas the adversary *can* purely relay lazy phases, note that if the adversary guesses wrongly at least one of the challenges, then authentication will fail. The same approach, but using quite expensive digital signatures, is used by the first distance-bounding construction of [13].

Terrorist Fraud. Terrorist fraud resistance is the strongest attack considered in standard security models for distance-bounding protocols. In the formulation of [4], terrorist fraud is an attack where the adversary breaks the soundness of the protocol with the dishonest prover's aid, such that the information forwarded by the prover: (a) maximises the adversary's winning probability, and (b) does not give the adversary any advantage for future attacks. Whereas this definition seems to capture the intuition, it is not very formal. The concept of advantage is not formally defined: as it stands, the definition suggests that a proof of terrorist fraud resistance must show that every adversary that wins with some probability in the presence of the dishonest prover wins with at most the same probability without the prover's aid. It is also unclear what kind of maximisation is implied by (a) and how one would go about proving that the adversary's winning probability is maximised. However, this definition of terrorist fraud resistance immediately implies mafia fraud resistance: indeed, if the protocol is sound, then the mafia fraud adversary in particular has a negligible advantage to break its soundness. Also, the terrorist adversary in the absence of the dishonest tag runs a mafia fraud game; therefore point (b) enforces mafia fraud resistance. We note that later work from the same authors, i.e. [5] seems to require information-theoretical hiding properties for the secret key shared between the reader and the tag. We note, however, that this seems too strong a restriction: intuitively, an adversary may learn some information about the key without gaining any advantage in future attacks.

By contrast, Definition 2.5 describes a very strong terrorist fraud notion (we refer to Chapter 5 for more relaxed notions of terrorist fraud), with very few restrictions on how the prover may help the adversary. Thus the adversary is allowed to query the dishonest prover at will during all the slow (lazy) phases of the protocol (but not during the time-critical phases). As we will show in Section 2.3, existing constructions in the literature that address the intuitive notion of terrorist fraud resistance (and which are presumably terrorist fraud resistant in the sense of [4], though this has not formally been proved), are *not* secure in our definition (see Sections 2.3.5 and 2.3.4, which assess the protocols of Reid et al. and the Swiss-Knife protocol due to Kim et al.).
The contradiction between the *claimed* terrorist fraud resistance of such schemes and the resistance we can *prove* they achieve may indicate that our definition is *too* strong. The main difficulty in achieving terrorist fraud resistance is to link authentication sessions such that once an authentication attempt succeeds, some inside information recovered by the adversary allows it to authenticate again. This is not a trivial task. One may argue, for example, that a protocol could be terrorist fraud resistant if, once a single authentication attempt is successful, the verifier will always accept future authentication attempts, regardless of the prover's behaviour. This, of course, would make the construction completely insecure, as an adversary could simply eavesdrop on a regular authentication attempt (since the tag is honest, the attempt would succeed), and then it could authenticate by default, breaking security. We argue that the correct definition of terrorist fraud resistance depends on the application where the distance-bounding protocol is deployed. In particular, in our definition, the dishonest prover is willing to leak some information about its secret key, as long as it does not enable the simulator to authenticate with equal probability as the adversary. Whereas we are able to relax this definition to a different notion, where the dishonest prover is no longer willing to hand out *any* information about the secret keys, see Chapter 5, our results still diverge from those achieved by Avoine et al. [4] (thus indicating that our formalisations are not formally equivalent).
In the literature, protocols addressing terrorist fraud generally use essentially two possible responses for every round (one response for a 0 challenge, and another response for the 1 challenge). If an adversary learns both challenges for all the time-critical rounds, then it also learns the secret key *sk* and can thus authenticate without the prover's consent. Therefore, the attack we described against the Hancke and Kuhn protocol [42] is no longer valid, as the dishonest prover can't prepare the adversary for *all* challenges without revealing the secret key. Whereas this idea seems intuitively sound, we point out that none of the constructions in the literature have been *proved* secure.

Distance Fraud. Avoine et al. [4] defines distance fraud as above, i.e. a dishonest, legitimate prover purports to be within the neighbourhood of the verifier. This definition relies on the fact that the verifier's clock will detect time of flight accurately: thus, in order to respond within the time limit, the prover has to respond to the verifier's challenges *before* the challenges are actually sent. This is exactly our approach here.
Note that generally speaking, for protocols where time-critical phases contain a single round, with the verifier's one-bit challenge and the corresponding one-bit response, the dishonest prover has a probability of $\frac{3}{4}$ to guess the correct response at every round. Such an adversary typically runs as follows: given a one-bit round response r_0 for a 0 challenge and a one-bit response r_1 for a 1 challenge (the prover knows r_0 and r_1), if $r_0 = r_1$, the prover forwards this value. Else, if $r_0 \neq r_1$, the prover always sends r_0. If the responses are pseudorandom, then $r_0 = r_1$ with probability $\frac{1}{2}$. With probability $\frac{1}{2}$, the two responses are *not* equal; however, in this case, the prover has a probability of $\frac{1}{2}$ to guess the 0 challenge. The adversary's total success probability is $\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}$ per round. Thus, the distance fraud resistance of any protocol is lower-bounded by $\left(\frac{3}{4}\right)^{N_c}$, where N_c is the number of time-critical phases (i.e. the adversary must respond correctly every time).
In the literature, distance fraud resistance is generally addressed by randomising the verifier's challenges, such that the prover cannot predict them and forward the adequate response. However, recent work shows that this is not enough to ensure distance fraud resistance, see [11]. An additional necessary condition is that the correct responses for the 0 and resp. 1 challenge should be distributed randomly (such that the prover's probability of guessing the correct answer without knowing the challenge is $\frac{1}{2}$). In particular, the responses for the 0 and resp. 1 challenges should seem random *even if the adversary knows the secret key*. This very subtle aspect of the distance fraud notion is exploited by Boureanu et al. [11], who show an attack in which the dishonest prover chooses a convenient "weak" nonce, thus obtaining very similar response strings for 0 and 1 challenges. This does not violate the security assumption of the underlying primitive (i.e. a pseudorandom function), since security is there defined *on the average*, i.e. for most inputs, and for adversaries who do not know the secret key.

Impersonation Security. Whereas Avoine et al. [4] classify impersonation security as the *total* probability that an adversary within the verifier's proximity authenticates, we refer strictly to the adversary's success during lazy rounds. Thus, the same protocol may have a higher impersonation security level in the framework due to Avoine et al. [4], than for our

framework; the additional security comes from the time-critical rounds and is upper bounded by the mafia fraud resistance of the protocol. In practice, the notion is achieved by having the prover submit a verification string, based on the secret key, during one lazy protocol phase. Generally, lazy-phase authentication precedes the time-critical steps, but the protocols due to Brands and Chaum [13] and the Swiss-Knife protocol due to Kim et al. [50] stipulate lazy phases of authentication after the time-critical steps as an elegant way of achieving maximal mafia fraud resistance per round, as well as impersonation security.

White-Box vs. Black-Box. Avoine et al. [4] consider both a black-box and a white-box model. In the white-box model, the prover (but not a MITM adversary) may tamper with the protocol execution. This enables more efficient distance fraud attacks (as the prover can choose an input that maximises the number of positions where the responses for the two challenges are equal, an aspect also described by [11]). The white-box model may also imply more efficient terrorist fraud attacks, though it is unclear whether the prover can *use* the additional information in practice, since this also reveals (for most constructions in the literature) more information about the secret key.
In our framework, we allow an arbitrary runtime for adversaries; however, the prover and verifier algorithms must be *efficient*, thus polynomial. Generally in cryptography, efficiency only implies (local) polynomial computation and communication runtime, without taking into account the time required by other parties; in this sense, the attack described by [4] against [42], where the dishonest prover first receives a nonce and then runs a potentially exponential search to maximise its success probability *before* responding with its own nonce, is also valid in our model, which is therefore white-box. If, however, the notion of efficiency is also required to include the verifier's *waiting time* in lazy phases, then the model is not fully white-box.
We also give the adversary runtime as an explicit parameter: thus, the adversary's powers are unlimited, though quantified. Such an adversary may run an extended version of the attack of [4] offline, *before* the protocol execution: then, the reader also has a polynomial waiting time. In particular, instead of waiting for the nonce and then searching (in exponential run-time) for input that maximises the distance fraud attack success, the dishonest prover can make this search for *every* possible verifier input. Then, once the nonce is sent, the adversary picks the most convenient input for that value.
From a practical point of view, however, the white-box attack —and any exponential attack— is not likely to run for distance-bounding protocols, where the primitives used (in particular the pseudorandom functions used in the computation) change frequently with time. What is more interesting to observe is the relationship between the security notions, both in the black- and in the white-box model. Avoine et al. [4] claim that, in the black-box model, mafia and terrorist fraud resistance are equivalent, whereas in the white box model, terrorist fraud resistance implies mafia fraud resistance, but not vice versa. However, these statements seem questionable, depending on the interpretation of the definitions.
It can be argued that the Brands and Chaum protocol [13] is mafia fraud resistant in the black-box model of Avoine et al. [4], but *not* terrorist fraud resistant. This protocol first runs N_c time-critical rounds, where the verifier sends random challenges and the prover sends random responses; authentication is then achieved in a lazy phase by computing a digital signature on all the challenges and responses. A terrorist fraud adversary can first send random responses to all the challenges sent by the verifier; then the adversary sends both challenges and responses to the dishonest prover (in polynomial time). Having received the digital signature (which gives the adversary only a negligible advantage in future attempts) from the dishonest prover, the terrorist adversary forwards this value to the verifier and authenticates with probability 1. However, without the dishonest prover, the adversary's probability of forging a signature is negligible.
In the white-box model, terrorist fraud resistance is claimed to imply mafia fraud resistance. As noted above, this follows directly from the soundness assumption and the terrorist fraud definition; however, if a protocol is *not* sound, then the success of mafia fraud adversaries may be close to 1; then *any* help given by the dishonest tag will not give the terrorist adversary *additional advantage* (the adversary already has a success probability close to 1).
Avoine et al. [4] also claim that terrorist fraud resistance implies distance fraud resistance, and not viceversa, both in the black- and in the white-box models. The idea here is that if a dishonest prover can commit to the correct responses of the time-critical rounds before receiving the challenges, the same dishonest prover may forward the responses to an adversary acting as proxy, forwarding them (from within the range) when challenged. The soundness of the protocol protects against further advantages for this adversary, as the adversary also sees responses of honest runs anyway, between the prover and verifier. This argument holds as long as the internal delay within the man-in-the-middle (MITM) is smaller than the prover's advantage in guessing the responses. Otherwise, the responses from the adversary may still arrive too late, whereas they would arrive on time without the MITM delay (like in distance fraud).
By contrast, we showed in Section 2.1.3 that the four properties: mafia, distance, and terrorist fraud resistance, as well as impersonation security, are independent. We stress that our results do *not* automatically invalidate the results of [4], since the terrorist fraud resistance defined in Definition 2.5 is different, and stronger, than the one due to [4]. However, we note that without rigorous proofs and precise formalisations, the relationships claimed by [4] may provide a false sense

of security, as e.g. protocols that might be interpreted as terrorist fraud resistant will be immediately assumed to be mafia fraud resistant.

2.3 Prominent Constructions in the Literature

In this section we analyse the distance-bounding properties of the following protocols: the Brands and Chaum protocol [13], the Hancke-Kuhn protocol [42], the Avoine-Tchamkerten scheme [6], an improved version of the protocol due to Kim and Avoine [49], and the reputedly terrorist fraud resistant constructions due to Reid et al. [70] and Kim et al. [50]. For each protocol, we (1) show concrete security bounds for the four attacks described above, thus (2) disproving claims of terrorist fraud resistance of the well-known protocols in [70] and [50]; claims of distance fraud resistance for the schemes in [6, 42, 49]; and claims of generic distance and mafia fraud resistance for the protocol due to Reid et al. [70]. Particularly interesting are the distance fraud attacks we show, which are direct consequences of the recent work of Boureanu et al. [11], and the key-learning attack we show against the scheme of Reid et al. We also (3) discuss the consequences of the contradictions we find between the authors' claims and our results, balancing the strength of the models we use against the notion we intuitively wish to achieve.

One of the main results of this section is (2). Firstly, we show that the two allegedly terrorist fraud resistant constructions in [50] and [70] are *not*, in fact, terrorist fraud resistant. In particular, for Reid et al.'s construction, a malicious prover could forward the adversary a part (a half) of the correct responses and let the adversary guess the other responses. Thus, the adversary authenticates with probability of about $(\frac{3}{4})$ per time-critical round, whereas the simulator (i.e. the adversary without the prover's support) may only authenticate with probability $(\frac{1}{2})$ per round. This attack can furthermore be scaled down so that it even holds if we introduce a threshold value for the simulator's success probability (this, however, reduces the adversary's advantage).

In the Swiss-Knife protocol, the tag can even hand over its secret key, as long as it does not forward the secret identifier (the simulator has then no way of guessing the identifier, and cannot authenticate).

Secondly, we show a so-called key-learning attack against one instantiation of the scheme due to Reid et al. [70], where the symmetric encryption scheme required by the protocol is instantiated as bit-wise XOR. This is a particular type of mafia fraud attack, where the adversary exploits the fact that the time-critical responses are related in order to learn a long-term secret, enabling it to authenticate. The attack has a special form, consisting of several sessions when the adversary mostly eavesdrops on honest prover-verifier communication, merely flipping a few bits, in order to learn the long-term secret. This attack is not covered by previous frameworks, which assume that the adversary is never able to relay more than a tolerated number of time-critical phases. Key-learning attacks concern mostly protocols where the time-critical responses are related, i.e. in the case of protocols aiming to thwart terrorist fraud. We take particular care to avoid such attacks when we show our own terrorist fraud resistant scheme in Chapter 5.

We discuss at length the consequences of our results, with particular emphasis on the terrorist fraud resistant notion captured in our framework, which we assess in view of the intuition behind terrorist fraud resistance. Our attack against the construction of Reid et al. raises some questions, as this protocol does attain some intuitive, though weak, notion of terrorist fraud resistance, which excludes attacks where partial, indirect secret key information is forwarded. We argue that, whereas our definition is very strong, the weaker, intuitive notion is very weak. Furthermore, no other concrete formalisation of this intuitive notion is known, as the definition in [4] is very informal (see section 2.2.2). Two open questions left in this chapter (and answered in Chapter 5) are: (1) can the intuition of terrorist fraud resistance be captured better by a relaxation of the definition in Section 2.1.2? and (2) can we, in fact, attain the strong notion considered above?

As described above, one important update of previous work [27, 32] comes as a consequence of recent work due to Boureanu et al. [11], who prove, essentially, that no protocol in the literature where the responses are computed by means of a pseudorandom function using (partial) input from the tag is distance-fraud secure. In particular, the tag can use "weak" input that ensures that the Hamming distance between the two response strings (i.e. the number of bits where the response strings differ) is small, or even 0.

Note that one easy fix to this flaw is to increase computation and communication complexity and require that both the prover and verifier check that the Hamming distance between the two responses is higher than a certain value, e.g. $\frac{1}{3}N_c$, where N_c represents the number of time-critical phases; if this is not the case, then the tag has to re-generate the nonce. Thus, in order to preserve the level of distance-fraud resistance claimed by the respective protocols, we need to run more time-critical phases depending on the value we choose as a threshold (a lower value increases the adversary's probability to win, whereas a high value increases the probability that an honest tag generates a nonce that produces sufficient entropy in the responses). It is left to future research to find a more efficient way of bypassing the attack of [11].

Very interestingly, protocols which attempt to thwart terrorist fraud attacks can also attain terrorist fraud resistance. The main difference here is that the two responses are inter-dependent, and not computed as part of the pseudorandom function

output. In particular, while one of the two time-critical phase responses is computed as part of the pseudorandom function output, the other is computed separately, usually in such a way that, given both time-critical responses yields the secret key. While distance-fraud resistance cannot be generically proved for the protocol due to Reid et al. [70][7], the Swiss-Knife protocol due to Kim et al. [50] *is* provably distance-fraud resistant. On the other hand, however, relating the responses makes the protocol vulnerable to key-learning attacks as described in [61].
Our analysis results are summarised in Table 9. In this table we show only the loose security bounds for the four attacks described above (ignoring the small terms). The precise quantification can be found in the following sections. The protocol presented in Section 2.3.4 is the modified version of the Kim and Avoine protocol in [49].

	Mafia	**Terror**	**Distance**	**Impersonation**
[13] [1]	$\left(\frac{1}{2}\right)^{N_c}$	$\times$	$\times$	$\left(\frac{1}{2}\right)^{N_c}$
[42]	$\left(\frac{3}{4}\right)^{N_c}$	$\times$	$\times$	$\times$
[6] [2]	$\frac{1}{2}(N_c+2)\left(\frac{1}{2}\right)^{N_c}$	$\times$	$\times$	$\left(\frac{1}{2}\right)^{\lvert V\rvert}$
[70]	$\times$[3]	$\times$	$\times$[3]	$\times$
[50]	$\left(\frac{1}{2}\right)^{N_c-T}$	$\times$	$\left(\frac{3}{4}\right)^{N_c-T}$	$\left(\frac{1}{2}\right)^{\lvert V\rvert}$
[27]	$\left(\frac{1}{2}\right)^{N_c}$	$\times$	$\times$	$\left(\frac{1}{2}\right)^{\lvert V\rvert}$

Figure 9: Distance Bounding at a glance. [1]This protocol uses expensive primitives. [2]This protocol requires exponential storage requirements. [3]This statement is generic: though the protocol does not attain provable security for arbitrary, general instantiations of the primitives involved, it may do so for particular instantiations. We denote by N_c the number of time-critical rounds, by T a tolerance level for faults, and by $|V|$ is the bit length of an authentication string V sent by the prover

In the following, we assess the security properties of various prominent distance-bounding protocols. As a brief preliminary, however, we review the more important security notions recurring in our proofs. As a matter of notation, we denote by out $\leftarrow \mathcal{A}^{\mathcal{O}(\cdot)^Q}(\mathsf{in})$ the fact that an algorithm $\mathcal{A}$ is given input in and may query an oracle $\mathcal{O}$ a total number Q of times, finally outputting an output out.

Pseudo-randomness. Pseudorandom functions (PRFs) are (publicly known) families of functions $F: \mathcal{K} \times D \to R$, where $\mathcal{K} = \{0,1\}^k$ is the space of keys, $D = \{0,1\}^\ell$ is the domain of the function, and $R = \{0,1\}^L$ is the range, for integers $k, \ell, L \geq 1$. Intuitively, pseudorandomness indicates the lack of correlation between the input and output of the function. More formally, the security of pseudorandom functions is defined in terms of a game, where an adversary receives a set of values which can be either the output of F for a fixed key $K \in \mathcal{K}$, or a truly random number. We quantify adversaries in terms of their runtime t and the number of queries Q which they make to the oracle $\mathcal{O}$.

Definition 2.13 *Let $F: \mathcal{K} \times D \to R$ be a family of functions as above. The* distinguishing *advantage* $\mathbf{Adv}^d_{\mathsf{PRF}}(\mathcal{A})$ *of an adversary $\mathcal{A}$, running in time against F is defined as:*

$$\mathbf{Adv}^d_{\mathsf{PRF}}(\mathcal{A}') = \left|\mathrm{Prob}\left[\,\mathsf{Exp}^{\mathsf{PRF}}_{\mathcal{A}} = 1\right] - \frac{1}{2}\right|,$$

where we define $\mathsf{Exp}^{\mathsf{PRF}}_{\mathcal{A}}$ *as follows. Let $\mathcal{O}_K(b,x)$ be an oracle which, on input a bit b and a value $x \in D$, and for a fixed, given key K, if $b = 1$ outputs $F(K,x)$ and else, if $b = 0$, outputs (consistent) random values r. Now the pseudorandomness experiment* $\mathsf{Exp}^{F}_{\mathcal{A}}$ *runs as follows.*

Experiment $\mathsf{Exp}^{\mathbf{F}}_{\mathcal{A}}$*:*
$K \leftarrow_R \mathcal{K}$
$b \leftarrow \{0,1\}$
$d \leftarrow \mathcal{A}^{\mathcal{O}_K(b,\cdot)^Q}$
The experiment outputs 1 *if* $d = b$ *and* 0 *otherwise.*

[7]Though the proof is not generic, i.e. it does not work for *every* symmetric encryption function, some *specific* instantiations of the scheme —e.g. one-time-pad encryption— might grant distance-fraud resistance.

Flavours of unforgeability. Most distance-bounding protocols in the literature use pseudorandom functions. However, the protocol due to Brands and Chaum [13] uses signature schemes instead. A signature scheme is defined as usual as a triplet of algorithms $\mathit{Sign} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ such that, on input a security parameter (in unary) 1^k, the key-generation algorithm SKg outputs a private-public key pair $(\mathit{sk}, \mathit{pk})$; on input a message m and the secret key *sk*, the signing algorithm $\mathsf{SSign}(\mathit{sk}, m)$ outputs a signature σ; and on input a message m, a signature σ, and the public key *pk*, the verification algorithm $\mathsf{SVf}(\mathit{pk}, m, \sigma)$ outputs a bit indicating whether the signature verifies (the output bit is then 1), or does not verify (the output bit is 0).
The security of signature schemes is usually defined in terms of unforgeability, which intuitively captures the fact that adversaries against the signature scheme should not be able to forge signatures for "fresh" message m even if they have the possibility to query correct signatures for arbitrary messages of its choice (but these messages should be different from m for the usual notion of existential unforgeability). More formally, let $\mathsf{Sign}(\mathit{sk}, m)$ be an oracle that, for a secret key *sk*, and a message m, outputs the signature $\mathsf{SSign}(\mathit{sk}, m)$. We quantify adversaries against the unforgeability of the signature scheme in terms of the runtime t and the number of queries Q to the Sign oracle, and we define unforgeability as follows.

Definition 2.14 *Let* $\mathsf{Sign} = (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ *be a signature scheme as above. Let* $\mathcal{A}$ *be an adversary against the unforgeability of* Sign*, running in time* t *and making at most* Q *queries to the signing oracle* Sign *(see the experiment below). Then the forging advantage of* $\mathcal{A}$ *is defined as*

$$\mathbf{Adv}^{\mathit{Unf}}_{\mathsf{Sign}}(\mathcal{A}) = \mathrm{Prob}\left[\,\mathsf{Exp}^{\mathsf{Sign}}_{\mathcal{A}}(1^k) = 1\right],$$

where $\mathsf{Exp}^{\mathsf{Sign}}_{\mathcal{A}}(1^k)$ *is defined as follows.*

Experiment $\mathsf{Exp}^{\mathsf{Sign}}_{\mathcal{A}}$*:*
$(\mathit{pk}, \mathit{sk}) \leftarrow \mathsf{SKg}(1^k)$
$(m, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathit{sk},\cdot)^Q}(\mathit{pk})$
The experiment outputs 1 *if (a)* $\mathsf{SVf}(\mathit{pk}, m, \sigma) = 1$ *and (b) the message* m *was not previously queried to* Sign.

For our security proof in the case of the Brands and Chaum protocol (see Section 2.3.1), we also require the property of *strong* unforgeability, where an adversary can also output a tuple (m, σ) with m previously queried to Sign, under the condition that σ is not the signature that the oracle Sign forwarded to the adversary. In fact, the definition is the same as in the case of existential unforgeability, but the security game is modified as follows.
Experiment $\mathsf{Exp}^{*\mathit{Sign}}_{\mathcal{A}}$:
$(\mathit{pk}, \mathit{sk}) \leftarrow \mathsf{SKg}(1^k)$
$(m, \sigma) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathit{sk},\cdot)^Q}(\mathit{pk})$
Let $L = \{m_i, \sigma_i\}_{i=1}^{Q}$ be the list of queries and responses to the signing oracle. The experiment outputs 1 if (a) $\mathsf{SVf}(\mathit{pk}, m, \sigma) = 1$ and (b) the tuple $(m, \sigma) \notin L$.

2.3.1 Brands and Chaum

In the Brands and Chaum construction [13], the reader and tag first exchange random bits in N_c time-critical phases, then finally signs the concatenation of these bits under the shared secret key *sk* — see Figure 10. Though in this case the lazy phases do not pre-date the time-critical phases, the concept of impersonation resistance is still applicable. We only require that the adversary can forward a fresh signature in a winning reader-adversary session, in the sense that the adversary has not seen it before. As the lazy phase has a single message, the only possibly relayed message here is the signature.
Intuitively, the time-critical phases of this protocol must ensure here that the reader-to-tag distance is no greater than the one associated with $t_{\max}$. Finally, the signature in the lazy phase must ensure that the bit exchange was done by a legitimate prover. This ensures both impersonation resistance and a measure of mafia fraud resistance. Lazy phases are susceptible to pure relay (which otherwise taints time-critical rounds), thus the signature yields neither terrorist, nor distance fraud resistance. Unlike most other distance bounding protocols, however, the tag's responses are fully random, thus the difficulty in mafia fraud attacks is not to answer the time-critical rounds, but rather to make the tag generate the correct signature at the end.
We formally state the properties of this protocol below, without considering any further fault tolerance. If the parameters $E_{\max}$ and $T_{\max}$ are considered, upper-bounding the number of erroneous transmissions and respectively the number of transmissions exceeding $t_{\max}$, the security bound for mafia fraud resistance drops by a factor $2^{T_{\max}}$. Note, however, that should *any* of the transmissions go wrong, the signature will not longer verify: for scenarios where transmissions are not reliable, the tag should append the values $R_1, \ldots, R_{N_c}, T_1, \ldots, T_{N_c}$ to the signature.

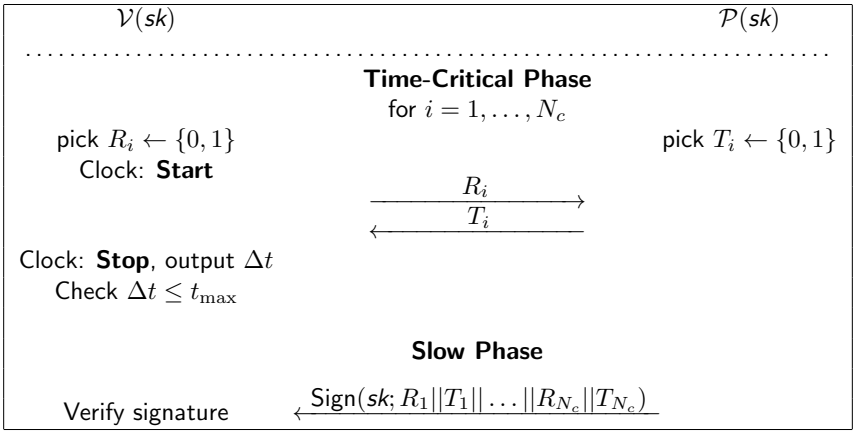

Figure 10: The Brands and Chaum protocol

Theorem 2.15 (Brands-Chaum Properties) *Let* Sign $= (\mathsf{SKg}, \mathsf{SSign}, \mathsf{SVf})$ *be the signature scheme used above. The Brands and Chaum protocol* DB *above has the following properties:*

- *It is not resistant to terrorist, nor to distance fraud.*
- *For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-impersonation adversary $\mathcal{A}$ against the Brands Chaum protocol, there exists an adversary $\mathcal{A}'$ against the strong unforgeability of* Sign *that runs in time t and requests at most $q_{\mathcal{P}} + q_{\text{OBS}}$ signatures, and that then outputs a valid forgery with an advantage $\mathbf{Adv}^{imp}_{\mathsf{DB}}(\mathcal{A})$ such that:*

$$\mathbf{Adv}^{imp}_{\mathsf{DB}}(\mathcal{A}) \leq \mathbf{Adv}^{strUnf}_{\mathsf{Sign}}(\mathcal{A}') + q_{\mathcal{V}}(q_{\mathcal{P}} + q_{\text{OBS}}) \cdot 2^{-N_c}.$$

- *For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-mafia-fraud adversary $\mathcal{A}$ against the Brands Chaum protocol, there exists an adversary $\mathcal{A}'$ against the unforgeability of* Sign *that runs in time t and requests at most $q_{\mathcal{P}} + q_{\text{OBS}}$ signatures, and that then outputs a valid forgery with an advantage $\mathbf{Adv}^{Unf}_{\mathsf{Sign}}(\mathcal{A}')$ such that:*

$$\mathbf{Adv}^{mafia}_{\mathsf{DB}}(\mathcal{A}) \leq \mathbf{Adv}^{Unf}_{\mathsf{Sign}}(\mathcal{A}') + q_{\mathcal{V}}(q_{\mathcal{P}} + q_{\text{OBS}}) \cdot 2^{-N_c} + q_{\mathcal{V}} \cdot 2^{-N_c}.$$

Proof. The second statement is easily proved. Consider a reader-adversary session sid where an adversary $\mathcal{A}$ successfully impersonates to the reader. Firstly, there is at most a probability of $(q_{\text{OBS}} + q_{\mathcal{P}}) \cdot 2^{-|N_c|}$ that the challenges in this verifier-adversary session coincide with the challenges R_i in a previous reader-tag or adversary-tag session. In this case, the adversary can replay the past session (including the signature received at the end), thus winning with probability 1. We account for this probability in each of the verifier-adversary sessions, and assume from now on that the string of challenges $\{R_1, \ldots, R_{N_c}\}$ is unique amongst all the past reader-tag and adversary-tag sessions.
We consider now a single reader-adversary session sid. The adversary can now open a parallel adversary-tag session sid$'$. The definition of impersonation security requires that the adversary does not forward lazy phase information between the two sessions. The adversary can, however, forward the time-critical phase information between the two sessions. We show that the adversary can now win with probability at most equal to the advantage of any adversary against the strong unforgeability of the signature scheme *Sign*. Indeed, we construct such an adversary $\mathcal{A}'$ which runs $\mathcal{A}$ in a black-box way. Every time $\mathcal{A}$ runs either a prover-verifier or resp. adversary-prover session, the adversary $\mathcal{A}'$ queries its signature oracle for the transcript of the protocol, and forwards the signature to $\mathcal{A}$, storing the value of $\mathsf{SSign}(sk; R_1||T_1|| \ldots ||R_{N_c}||T_{N_c})$. Finally, assuming that $\mathcal{A}$ succeeds in impersonating the prover in a session sid, the adversary $\mathcal{A}'$ forwards the tuple consisting of the time-critical transcript of the protocol, together with the signature used by $\mathcal{A}$ in its successful attempt, to the challenger in the strong unforgeability game. In case the adversary has relayed the time-critical transcript, the adversary $\mathcal{A}$ only wins if it forwards a different signature for the same message, i.e. we must require strong unforgeability. If the impersonation adversary $\mathcal{A}$ has an advantage $\mathbf{Adv}^{\text{imp}}_{\mathsf{DB}}(\mathcal{A})$, then the unforgeability adversary $\mathcal{A}'$ has a probability at least as large to succeed in its attempt. Accounting for $q_{\mathcal{V}}$ possible replays, we obtain the specified bound.
The first statement also follows easily: the adversary $\mathcal{A}$ against distance fraud simply commits to randomly chosen values of T_i, then computes the (correct) signature with its own secret key *sk*, and succeeds with probability 1. For terrorist

fraud, the adversary trivially generates random time-critical responses, then forwards these values and the values of R_i to the dishonest prover, and receives the signature. The success probability of this adversary is 1. However, the simulator's success probability is upper bounded by the probability that it can forge a signature for a fresh session. For an unforgeable signature scheme, the simulator's success probability is negligible, and thus the scheme is not terrorist fraud resistant.
We now prove the last statement. Consider the adversary $\mathcal{A}$ mounting a mafia fraud attack. Consider the reader-adversary session sid where $\mathcal{A}$ successfully authenticates to the reader. As in the impersonation fraud proof, there is a probability of $(q_{\mathcal{P}} + q_{\text{OBS}}) \cdot 2^{-N_c}$ that the challenges R_i in sid are all identical to the fast phase values R_i^* in one of these sessions. In this case, the adversary can simply forward the observed/received values T_i^* to the reader in session sid; at the end, the adversary forwards the observed/received value of the signature. We now assume that the verifier-adversary session sid is the only session the adversary has access to. The adversary can start a concurrent adversary-tag session sid*. We call a round successful if the reader input R_i and the response T_i in round i of sid are the same as the adversary input R_i^* and the response T_i^* in sid*, *without* pure relay, i.e. the adversary must send R_i^* in sid* before receiving R_i in sid or it must guess T_i in advance, before receiving T_i^*.
We first show that, except with negligible probability, the adversary cannot win unless all time-critical rounds are successful. Indeed if one round is not successful (i.e. R_i^* in sid is different from R_i in sid*) and yet $\mathcal{A}$ authenticates to the reader in sid, we can build an adversary against the unforgeability of the signature scheme. This adversary's forgery is the message m containing the concatenation of all the queries and responses in session sid and the signature that $\mathcal{A}$ forwards in sid. The signature must be correct – else $\mathcal{A}$ cannot authenticate —and the message is fresh, as at least in round i the value of $R_i||T_i$ is fresh. Thus the probability that $\mathcal{A}$ wins without being successful in each round is $\mathbf{Adv}^{\mathsf{Unf}}_{\mathit{Sign}}(\mathcal{A}')$.
If the adversary is successful in every round, then it wins with probability 1, since in this case the time-critical transcripts are identical between sid and sid* and the adversary can just relay the lazy-phase response. However, since the challenges and responses are independent, the probability that the adversary is successful in every round is $\left(\frac{1}{2}\right)^N_c$ for each of the reader-adversary sessions. Accounting for all reader-adversary sessions, we obtain the bound above. □

2.3.2 Hancke and Kuhn

One weakness of the Brands and Chaum construction is the use of the digital signature, which cannot be easily implemented on RFID tags. Also, [13] offers no distance fraud resistance, as the tag's responses are not challenge specific and can be sent in advance. The protocol due to Hancke and Kuhn uses a pseudorandom function (PRF) instead of the signature scheme, implemented as HMAC. It also gains distance fraud resistance at the cost of a lower mafia fraud resistance. The protocol consists of a lazy phase, where the parties exchange nonces and pre-compute an HMAC value, which is divided in a left and a right half, and N_c time-critical phases where the reader forwards a random bit and the tag responds with a bit either from the left or right half of the PRF output. This is also depicted in Figure 11.

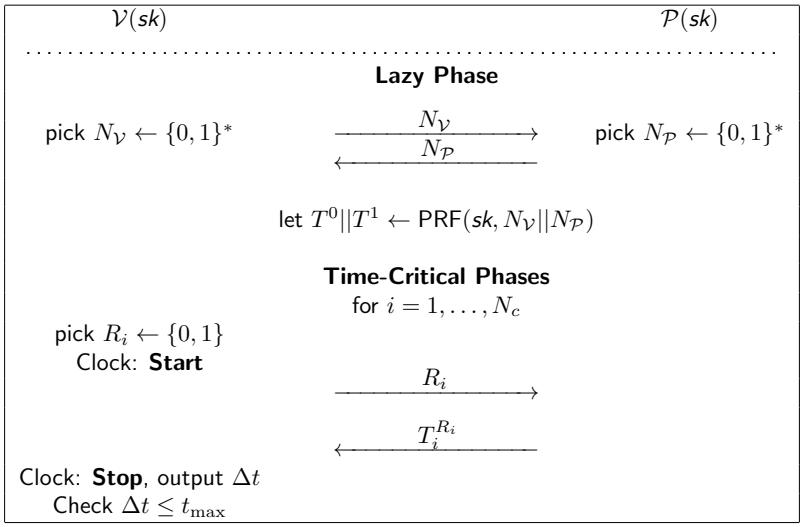

Figure 11: The Hancke and Kuhn protocol

The PRF output is $2N_c$ bits, and the left and right halves of this output have equal length. Though this protocol is more easily implementable on RFID, the lack of reader authentication leads to a decreased mafia fraud resistance; also, in view of [11], the fact that the tag chooses part of the PRF input (and may do so dishonestly) disables distance fraud resistance. As in [13], there is no impersonation resistance. We formally state these properties below.

Theorem 2.16 (Hancke-Kuhn Properties) *Let* DB *be the distance-bounding authentication scheme in Figure 11 with parameters* $(t_{\max}, N_c)$*. This scheme has the following properties:*

- *It is not impersonation resistant, distance fraud, nor resistant to terrorist fraud.*
- *For any* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-mafia-fraud adversary* $\mathcal{A}$ *against the scheme there exists a* (t', q')*-distinguisher* $\mathcal{A}'$ *against* PRF *(where* $\mathcal{A}'$ *runs in time* $t' = t + O(n)$ *and makes at most* $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$ *queries) such that*

$$\mathbf{Adv}^{\textit{mafia}}_{\mathsf{DB}}(\mathcal{A}) \quad \leq \quad q_{\mathcal{V}} \cdot \left(\tfrac{3}{4}\right)^{N_c} + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \mathbf{Adv}^{d}_{\mathsf{PRF}}(\mathcal{A}') + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}.$$

Proof. The first statement follows easily: an impersonation adversary which simply generates a random nonce during the single lazy phase of this protocol wins with probability 1 as there is no authentication during this lazy phase. For terrorist fraud resistance, a malicious tag equips the adversary during the lazy phase of session sid with the output $T^0||T^1$ of the PRF for nonces $N_{\mathcal{V}}$ and $N_{\mathcal{P}}$. This adversary succeeds with probability 1, as it can answer the queries of the honest reader in sid as though it were the legitimate tag. Now consider a simulator attempting to authenticate with the data obtained from the adversary. However, the simulator's session is fresh and so is the reader's nonce. Thus either the PRF output is different, or we can find a collision in the PRF. Finally, the insecurity of this protocol against a distance-fraud resistance attack follows from [11], i.e. the tag will simply choose a "weak" nonce $N_{\mathcal{P}}$ such that $T^0 = T^1$. As the protocol uses PRF in a black-box way, the protocol has to work with *any* pseudorandom function. However, pseudorandom only guarantees average-case randomness of its output; in other words, it is possible that a few inputs yield outputs that do not resemble purely-random values.

The proof of the last statement consists of the following high-level steps:

1. Show that one can safely replace the PRF runs of honest parties by picking independent random strings $T^0||T^1$ for each new nonce pair $(N_{\mathcal{V}}, N_{\mathcal{P}})$. Note that the value of M, which is sent in clear, has no influence on this step.
2. Show that nonce pairs are (almost) unique, except for possibly one adversary-tag session sid* having the same nonce pair as a reader-adversary session sid (here the adversary relays the nonces between sessions).
3. Bound the probability that the adversary passes the time-critical phases for at most one adversary-tag interaction.

For the first step we claim that replacing the PRF-values by random (but consistent) values can at most decrease the adversary's success probability by the distinguishing advantage for PRF. This can be seen easily by construction adversary $\mathcal{A}'$ against PRF via black-box simulation of $\mathcal{A}$, each time applying the random or pseudorandom oracle to nonce pairs on behalf of the honest parties. Finally, $\mathcal{A}'$ checks if $\mathcal{A}$ succeeds in some reader-adversary session and outputs 1 if this happens. The distinguishing advantage of $\mathcal{A}'$ then corresponds to the decrease of the success probability of $\mathcal{A}$ when switching to random values $T^0||T^1$.

Next consider the adversary $\mathcal{A}$ mounting a mafia fraud attack and all the pairs of nonces appearing in the attack. Assume that there exist two sessions (between adversary and tag or reader, or between both honest parties) with the same pair $(N_{\mathcal{V}}, N_{\mathcal{P}})$. Then we claim that this can only be a reader-adversary session and an adversary-tag session, except with probability

$$\binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}.$$

This holds as for each two executions for the reader resp. tag the nonce of this party is picked at random. If three identical nonce pairs appear in some executions then two of them are either in the at most $q_{\mathcal{V}} + q_{\text{OBS}}$ executions with the reader, or in the $q_{\mathcal{P}}$ executions with the tag. Such collisions occur with the above probability.

Declare $\mathcal{A}$ to lose if a collision appears, decreasing its success probability by this negligible term, but allowing us to consider collision-free executions. In particular, except for the matching session, all values $T^0||T^1$ in the attack are independent.

Now consider a reader-adversary session sid in which $\mathcal{A}$ successfully impersonates to $\mathcal{V}$. By assumption the same nonce pair appears in at most one other adversary-tag session. If there exists a (unique) matching adversary-tag session sid* then we claim that this session taints sid with high probability (if there is no such session, we have the case below, where the adversary does not take advantage of a matching session). Since for this protocol it holds that $T_{\max} = 1$, this invalidates session sid. Suppose, to the contrary, that the matching session sid* taints no time-critical phase in sid.

Consider an untainted time-critical phase of sid where $\mathcal{V}$ sends $R_i = b$ and expects T_i^b. The adversary has thus successfully passed the first $i-1$ time-critical phases and can choose to do one of the following in the i-th phase:

The Go-Early Strategy. In session sid* the adversary has sent some bit R_i^* to $\mathcal{P}$ before having received R_i (i.e., clock(sid, $i+2$) > clock(sid*, $i+2$) in the notation of Section 2.1.2). Then, since R_i is random and independent of all other data, the probability of $R_i^* \neq R_i$ is $\frac{1}{2}$, in which case $\mathcal{A}$ does not receive T_i^* in sid* and can only guess the value T_i in sid. If $b = R_i = R_i^*$, however, the adversary returns the correct reply T_i^b with probability 1.

The Go-Late Strategy. In session sid the adversary replies to R_i with some T_i^* before receiving $(T_i^b)^*$ in session sid* (i.e., clock(sid, $i+3$) < clock(sid*, $i+3$)). Now $\mathcal{A}$ succeeds only with probability $\frac{1}{2}$ for this phase.

The Modify-it Strategy. The adversary schedules the message such that it receives R_i in sid, sends some $R_i^* = b$ in sid*, receives $(T_i^b)^*$ in sid*, and forwards some T_i^* in sid. Hence, the scheduling corresponds to a pure relay attack, but $R_i \neq R_i^*$ or $T_i^* \neq (T_i^b)^*$. If $b = R_i^*$ is wrong then $(T_i^b)^*$ is actually never sent by $\mathcal{P}$ in sid* and the adversary can thus only guess T_i^* with probability $\frac{1}{2}$; if $b = R_i = R_i^*$ then $T_i^* \neq (T_i^b)^*$ makes the reader reject.

The Taint-it Strategy. The adversary taints this phase of sid through sid*. This is equivalent here to losing in sid.

The Taint-it strategy may be ignored, as it disables sid. The Go-Late and Modify-it strategies both succeed with probability at most $\frac{1}{2}$. The Go-Early strategy succeeds with probability $\frac{3}{4}$. As all rounds are independent, and taking into account the $q_{\mathcal{V}}$ trials, this gives the claimed bound. □

2.3.3 Avoine and Tchamkerten

As noted in the previous section, the Hancke and Kuhn construction has interdependent challenges, allowing the adversary to run a complete number of time-critical exchanges with the tag before attempting to authenticate to the reader. The Avoine and Tchamkerten protocol tries to correct this flaw by providing some reader authentication. The main idea is to store the secret key in one (or more) binary tree(s); the challenges are inter-related, with the responses forming paths from the root to the leaves.

Consider now an adversary impersonating a reader in an adversary-tag session in a Go-Early strategy as in the proof of Theorem 2.16. For [42], an adversary can choose challenges R_i^* for each round $i = 1, \ldots, N_c$ and receive responses T_i^* from the tag, having a 50% probability to have queried the honest tag with the correct $R_i^* = R_i$; for these queries, the adversary will know the correct response $T_i = T_i^*$. On the other hand, once an adversary makes an incorrect guess for some R_i^*, none of the future responses will be the correct ones.

This is also depicted in Figure 12.

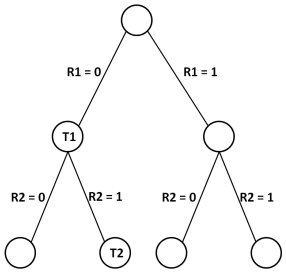

Figure 12: The Avoine-Tchamkerten protocol where the reader sends challenges $R_1 = 0$, $R_2 = 1$

Note that this protocol requires at least $2^{N_c+1} - 2$ potential response bits; at the expense of security, one could consider more than just one tree. We have, however, lazy phase authentication, leading to a total PRF output of $m + 2^{N_c+1} - 2$ bits for acceptably large m. The first m bits of the PRF output are denoted M; the last $2^{N_c+1} - 2$ bits are denoted T and stored in a tree as follows: from the top downwards and from left to right, each node is labelled with an output bit. The edge between each node and its left child is labelled 0 and the edge to the right child is labelled 1. We denote by Node$(R_1, \ldots, R_i)$ the label of the node that has the path $R_1, \ldots, R_i$ from the root. With this notation, the protocol runs as shown in Figure 13. Note in particular that the responses T_i do not correspond directly to the bits of T, but are rather chosen from amongst the bits of T according to the path $R_1, \ldots, R_i$.

Intuitively, the m-bit value V offers impersonation resistance. While the inter-dependency between the challenges of the reader increases the protocol's mafia fraud resistance, this is still not optimal, and it requires a great deal of storage. We formalise the concrete security of this construction below.

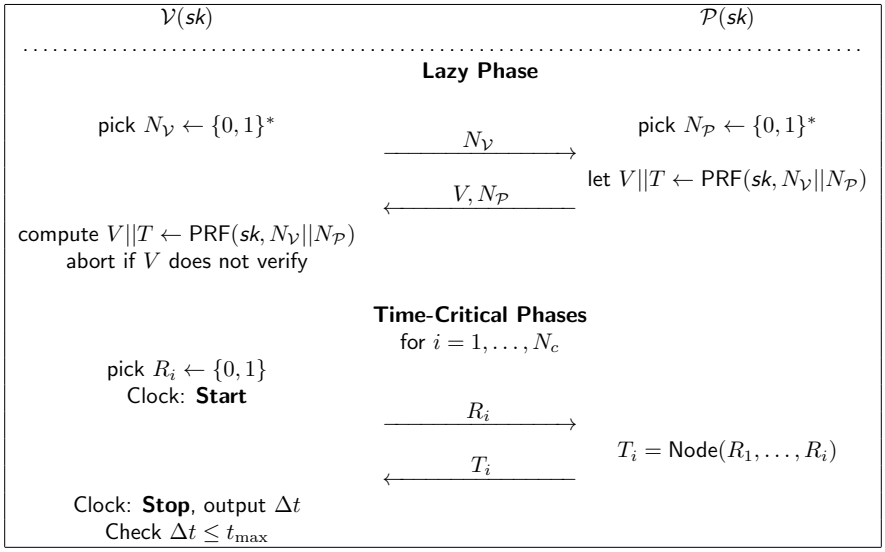

Figure 13: The Avoine and Tchamkerten protocol

Theorem 2.17 (Avoine-Tchamkerten Properties) *Let* DB *be the distance-bounding authentication scheme in Figure 13 with parameters* $(t_{\max}, N_c)$*. This scheme has the following properties:*

- *It is not resistant to terrorist fraud, nor to distance fraud.*
- *For any* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-mafia-fraud adversary* $\mathcal{A}$ *against the scheme there exists a* (t', q')*-distinguisher* $\mathcal{A}'$ *against* PRF *(where* $t' = t + O(n)$ *and* $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$*) such that*

$$\mathbf{Adv}^{mafia}_{\mathsf{DB}}(\mathcal{A}) \quad \leq \quad \tfrac{1}{2} q_{\mathcal{V}}[N_c + 2] \cdot 2^{-N_c} + \mathbf{Adv}^{d}_{\mathsf{PRF}}(\mathcal{A}') + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}.$$

- *For any* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-impersonation adversary* $\mathcal{A}$ *against the scheme there exists a* (t', q')*-distinguisher* $\mathcal{A}'$ *against* PRF *(where* $t' = t + O(n)$ *and* $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$*) such that*

$$\mathbf{Adv}^{imp}_{\mathsf{DB}}(\mathcal{A}) \quad \leq \quad q_{\mathcal{V}} \cdot 2^{-|V|} + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \mathbf{Adv}^{d}_{\mathsf{PRF}}(\mathcal{A}') + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}.$$

Proof. The proofs of the first statement is identical to those of the Hancke-Kuhn protocol, as shown in Theorem 2.16. For mafia fraud resistance, we change the Hancke-Kuhn proof as follows: for each round, we denote by pass_j the event that an adversary guesses the correct R_i in a Go-Early attack. Again the Go-Early strategy is the most effective, with a success probability given by the iterative expression below:

$$\mathrm{Prob}\left[\bigwedge\nolimits_{j=i}^{N_c} \mathsf{pass}_j \;\middle|\; \bigwedge\nolimits_{j=1}^{i-1} \mathsf{pass}_j\right] \leq \tfrac{1}{2} \cdot \tfrac{1}{2}^{N_c - i + 1} + \frac{1}{2} \cdot \mathrm{Prob}\left[\bigwedge\nolimits_{j=i+1}^{N_c} \mathsf{pass}_j \;\middle|\; \bigwedge\nolimits_{j=1}^{i} \mathsf{pass}_j\right].$$

After summing up and iterating, we have the bound above.
Finally, impersonation security follows similarly as the mafia fraud resistance. We first account for nonce-collisions between authentication sessions, as in Theorem 2.16. The impersonation adversary's advantage is now given by the advantage of the distinguisher $\mathcal{A}'$ and the probability that the adversary knows the output of the PRF for the successful reader-adversary session sid (which only happens if it has seen, resp. generated the authentication value V in a reader-tag, resp. adversary-tag session). □

2.3.4 The Improved Kim and Avoine Protocol

The scheme of Kim and Avoine in [49] is mafia and distance fraud resistant. We tweak it to add impersonation security, provide for noisy channels as in section 2.1, then prove its security properties in our framework. The proof relies on the fact that the nonce pairs exchanged in each run are quasi unique; also for any efficient adversary $\mathcal{A}'$ the advantage $\mathbf{Adv}^{\mathsf{dist}}_{\mathsf{PRF}}(\mathcal{A}')$ of distinguishing a pseudorandom function from a truly random one is small.
In particular, this protocol (and the underlying protocol of Kim and Avoine [49]) also achieves a dependency of the challenges, but by different means than the previously-presented protocol of Avoine and Tchamkerten see Section 2.3.3. In particular, the reader here *pre-computes* part of the challenges for time-critical rounds: and in these time-critical rounds, the tag can, as in the previous protocol, detect an attempted MITM attack. In particular, the output of the PRF in the previous protocols now also contains strings C and D, whose length is N_c, i.e. the number of time-critical rounds. For each time-critical round, if the corresponding bit of C is 0, the challenge will be randomly generated, and if the bit is 1, then the challenge is chosen from amongst the bits of D. The tag always responds with either the response v^0 or v^1, depending on the value of the challenge.

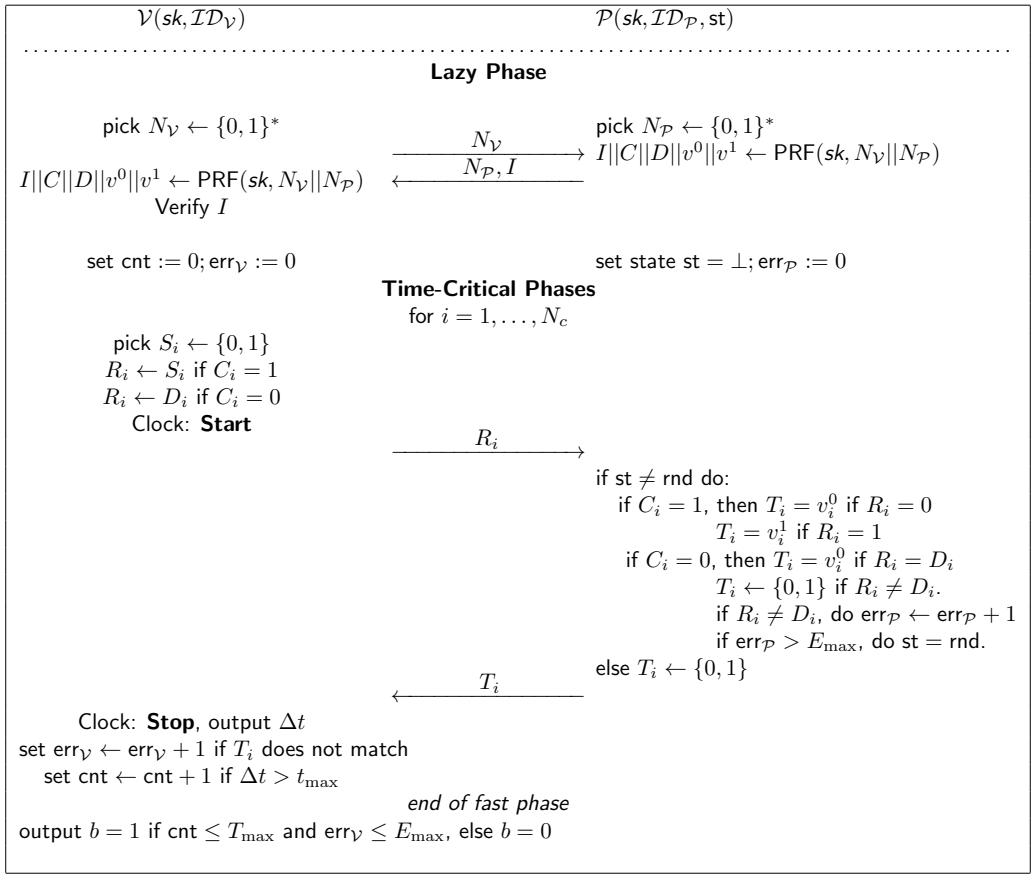

Figure 14: Enhanced Kim-Avoine protocol

For a single impersonation attempt and $T_{\max} = E_{\max} = 0$ we have up to small terms the (almost optimal) bound $\frac{1}{2}(N_c + 2) \cdot 2^{-N_c}$ for mafia fraud resistance.

Theorem 2.18 (Security Properties) *The distance-bounding authentication scheme* DB *in Figure 14 with parameters* $(T_{\max}, t_{\max}, E_{\max}, N_c)$ *has the following properties:*

- *It is not terrorist-fraud resistant, nor distance-fraud resistant.*

- *For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-impersonation adversary $\mathcal{A}$ against* DB *there exists a (t', q')-distinguisher $\mathcal{A}'$ against* PRF *(with $t' = t + O(n)$ and $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$) such that,*

$$\mathbf{Adv}^{imp}_{\mathsf{DB}}(\mathcal{A}) \leq q_{\mathcal{V}} \cdot 2^{-|I|} + \mathbf{Adv}^{dist}_{\mathsf{PRF}}(\mathcal{A}') + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}.$$

- *For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-mafia-fraud adversary $\mathcal{A}$ against* DB *there exists a (t', q')-distinguisher $\mathcal{A}'$ against* PRF *(where $t' = t + O(n)$ and $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$) such that, for $N_t = T_{\max} + 2E_{\max}$,*

$$\begin{aligned}\mathbf{Adv}^{mafia}_{\mathsf{DB}}(\mathcal{A}) \quad \leq \quad & \tfrac{5}{8} \cdot q_{\mathcal{V}} \binom{N_c}{N_t} \cdot (N_c - N_t + 2) \cdot 2^{-(N_c - N_t)} + \mathbf{Adv}^{dist}_{\mathsf{PRF}}(\mathcal{A}') \\ & + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}.\end{aligned}$$

Proof. The protocol is not terrorist-fraud resistant: $\mathcal{P}'$ can forward adversary $\mathcal{A}$ the value $I||C||S||v^0||v^1$. Now $\mathcal{A}$ authenticates successfully; a simulator can't authenticate, however, as a fresh session has new nonces in the lazy phase. The proof of distance-fraud insecurity is as in the previous proofs, as the protocol is vulnerable to the attack of Boureanu et al. [11].
As in Theorem 2.16, we prove mafia-fraud resistance as follows:

1. (1) Show that we can safely replace the honest parties' PRF output by independent random values $I||C||D||v^0||v^1$ for new nonces $(N_{\mathcal{V}}, N_{\mathcal{P}})$;
2. (2) Show quasi-uniqueness of nonce pairs except in 1 adversary-tag session and one reader-adversary session s.t. $\mathcal{A}$ relays the nonces;
3. (3) Bound $\mathcal{A}$'s winning probability in time-critical phases for at most one adversary-tag interaction.

For the first step, replacing the PRF-values by random (but consistent) values can at most decrease the adversary's success probability by the distinguishing advantage for PRF. Next consider the adversary $\mathcal{A}$ mounting a mafia fraud attack and all the pairs of nonces appearing the attack. Assume that there exist two sessions (between adversary and tag or reader, or between both honest parties) with the same pair $(N_{\mathcal{V}}, N_{\mathcal{P}})$. Then we claim that this can only be a reader-adversary session and an adversary-tag session, except with probability

$$\binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}.$$

This holds as for each two executions for the reader resp. tag the nonce of this party is picked at random. If there were three identical nonce pairs in some executions then two of them would be either in the at most $q_{\mathcal{V}} + q_{\text{OBS}}$ executions with the reader, or in the $q_{\mathcal{P}}$ executions with the tag. Such collisions can only occur with the above probability.
Declare the adversary now to lose if such a collision appears elsewhere, decreasing $\mathcal{A}$'s success probability only be the above negligible term, but allowing us to consider such collision-free executions from now on. In particular, except for the matching session all other values $I||C||D||v^0||v^1$ appearing in the attack are independent.
Let sid be a reader-adversary session where $\mathcal{A}$ successfully impersonates to $\mathcal{V}$. By assumption at most one other adversary-tag session sid* has the same nonce pair. If sid* exists, it taints sid with high probability (if sid* doesn't exist, $\mathcal{A}$ can't benefit from sid*). Suppose now that sid* taints at most $T_{\max}$ time-critical phases of sid. Assume for the moment that $E_{\max} = 0$; we make provisions for $E_{\max} > 0$ later.
Consider an untainted time-critical phase of sid where $\mathcal{V}$ sends R_i and expects T_i, i.e. assume $\mathcal{A}$ successfully passed the first $i-1$ time-critical phases. There are four strategies for the adversary in this i-th phase:

Go-Early. In session sid* $\mathcal{A}$ sends bit R_i^* to $\mathcal{P}$ before receiving R_i (i.e., clock(sid, $i+2$) > clock(sid*, $i+2$)). As R_i is random and independently chosen, $R_i^* \neq R_i$ w.p. $\frac{1}{2}$ —then $\mathcal{A}$ doesn't receive T_i in sid* and must guess T_i in sid. Also, session sid* becomes invalid with probability $\frac{1}{4}$.

Go-Late. In session sid, $\mathcal{A}$ replies to R_i with T_i before receiving T_i^* in session sid* (clock(sid, $i+3$) < clock(sid*, $i+3$)). Now $\mathcal{A}$ wins the phase w.p. $\frac{1}{2}$.

Modify-it. $\mathcal{A}$ receives R_i in sid, sends R_i^* in sid*, gets T_i^* in sid*, and forwards T_i in sid. This scheduling is pure relay, but $R_i \neq R_i^*$ or $T_i \neq T_i^*$. If R_i^* is wrong then T_i^* was never sent by $\mathcal{P}$ in sid* and $\mathcal{A}$ can only guess T_i w.p. $\frac{1}{2}$; if $R_i = R_i^*$ then $T_i \neq T_i^*$ makes the reader reject.

TAINT-IT. The adversary taints this phase of sid through sid*.

Tainting the phase makes $\mathcal{V}$ accept with probability 1, deducting 1 from the remaining taintable phases. The Go-Late and Modify-it Strategy both succeed w.p. at most $\frac{1}{2}$. Go-Early succeeds w.p. $\frac{3}{4}$, inactivating sid* w.p. $\frac{1}{2}$. Assume that $\mathcal{A}$ taints the last $T_{\max}$ time-critical phases (else we renumber the phases). For the other $P := N_c - T_{\max}$ phases let pass_i denote the event that $\mathcal{A}$ passes phase i of sid. We have

$$\mathrm{Prob}\Big[\bigwedge\nolimits_{j=i}^{P}\mathsf{pass}_j \;\Big|\; \bigwedge\nolimits_{j=1}^{i-1}\mathsf{pass}_j\Big] \le \frac{5}{8}\cdot\mathrm{Prob}\Big[\bigwedge\nolimits_{j=i+1}^{P}\mathsf{pass}_j \;\Big|\; \bigwedge\nolimits_{j=1}^{i}\mathsf{pass}_j\Big] + \tfrac{1}{2}\cdot\tfrac{1}{2}\cdot 2^{-P+i+1}$$

The first term captures the success of Go-Late, Modify-It, and correct Go-Early-prediction. The second term covers incorrect Go-Early prediction (w.p. $\frac{1}{4}$); now sid* is inactivated, and $\mathcal{A}$ must guess T_i for this and the next $P-i-1$ rounds (the responses are independent). Expanding the probabilities we obtain

$$\mathrm{Prob}\Big[\bigwedge\nolimits_{j=1}^{P}\mathsf{pass}_j\Big] \le 2^{-P} + \sum_{j=0}^{P-1}\frac{5}{8}\cdot 2^{-j}\cdot 2^{-P+j} = \tfrac{5}{8}\cdot(P+2)\cdot 2^{-P}.$$

We sum over $q_{\mathcal{V}}$ reader-adversary sessions, distribute $T_{\max}+E_{\max}$ "jokers" on the reader side and $E_{\max}$ on the tag side, and obtain the claimed bound.

For impersonation security, the only way to generate colliding nonce pairs (and produce authentication string I) is by lazy phase relay, which is an invalid impersonation attack. For distinct nonce pairs, the probability that $\mathcal{A}$ sends a correct I in a reader-adversary session is: $q_{\mathcal{V}}\cdot 2^{-|I|}$ plus the distinguishing advantage for the PRF plus the probability of colliding nonces.

□

2.3.5 Reid et al.

The construction in [6] has better mafia fraud resistance (but greater data storage) than [42]. The scheme in [6] is additionally impersonation resistant. However, neither scheme is terrorist fraud resistant, nor distance-fraud resistant. We now analyse the protocol due to Reid et al. [70], which adds a symmetric encryption scheme to the Hancke and Kuhn construction (the authors suggest a one-time-pad xor operation[8]).

Thus, this protocol inherits the lack of impersonation resistance of [42], as well as its vulnerability to Go-Early mafia fraud attackers. In fact, due to the inter-dependence of the two response strings, the protocol is also vulnerable to a different type of mafia fraud attack, called key-learning (see Chapter 4); in particular, it is impossible to generically prove mafia fraud resistance for this protocol. If the encryption scheme is instantiated as the one-time-pad, i.e. a xor operation, then the protocol is *not* mafia fraud resistant. However, we can also not build a generic mafia fraud attack against this scheme; in other words, it may be possible to instantiate the encryption scheme in such a way as to ensure mafia fraud resistance. We describe the attack against the xor instantiation; key-learning attacks in general are described in more detail in Chapter 4. Interestingly enough, distance-fraud resistance *also* can't be generically proved or disproved; this property might be achieved for some implementations of the symmetric encryption scheme, but not for others, since the attack of Boureanu et al. [11] cannot be extended to arbitrary schemes.

We first describe this protocol. In their paper [70] use a so-called key derivation function denoted KDF which can be viewed as a PRF for the sake of simplicity. We thus denote it PRF as for the construction due to Hancke and Kuhn. Furthermore, Reid et al. consider a symmetric IND-CPA encryption scheme denoted $\mathcal{E}$. To this scheme they associate a symmetric ephemeral secret key eph and a long term secret key *sk*. The notation $\mathcal{E}_{\mathsf{eph}}(sk)$ denotes the encryption under eph of the plaintext *sk*. We denote the corresponding decryption process by $\mathcal{D}$. Furthermore, Reid et al. [70] associate to the tag and reader some public identities $\mathcal{ID}_A, \mathcal{ID}_B$.

The main idea here is that both reader and tag compute a symmetric ephemeral key eph as the output of PRF, and then they use eph to encrypt the long-term secret key *sk* with $\mathcal{E}$. For each time-critical round, the reader challenges the tag with a random bit, and the tag responds with either a bit of the encrypted secret key or a bit of eph. Intuitively, terrorist fraud resistance results from the fact that the adversary requires either the long term secret *sk*(which can be later be used by the simulator to authenticate) or both eph and the encryption of *sk* (which can be used by the simulator to compute $\mathcal{D}_{\mathsf{eph}}(sk)$, thus also obtaining the secret *sk*). The scheme is depicted in Figure 15.

Before stating the security properties we note that the informal definition of a successful terrorist fraud attack is that the adversary can authenticate if aided by a malicious tag in a particular session, but cannot authenticate *without* such

[8]We note that Reid et al. also suggest different versions of this protocol, where the symmetric encryption is done differently; our analysis here applies equally to generic ways of implementing the encryption scheme.

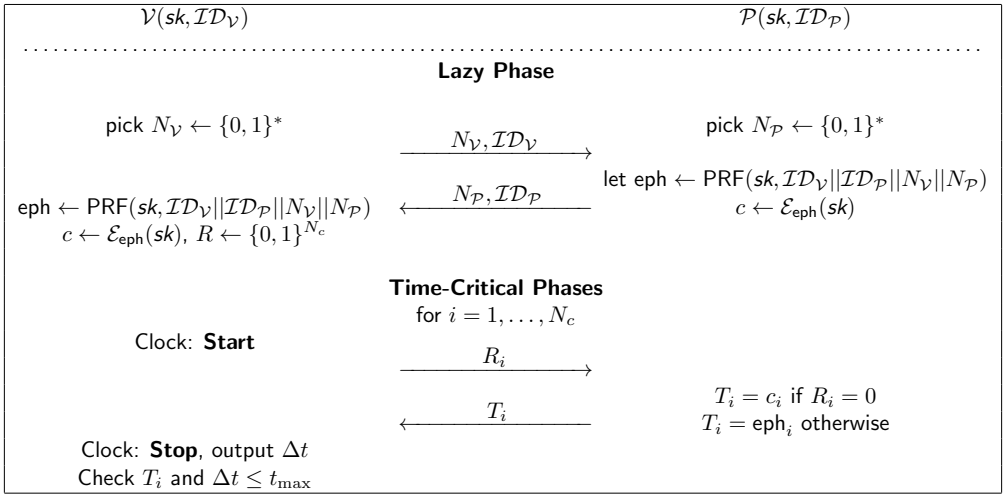

Figure 15: The Reid et al. protocol

additional aid. However, this informal definition is deceptive: many protocols claiming to be terrorist fraud resistant are actually resistant to the very restrictive requirement that if the adversary authenticates when aided by the malicious tag, then this facilitates, however little, future authentication sessions. By tying the knowledge (or accurate guessing) of both responses to the secret key, protocols can attain this form of terrorist fraud resistance. In other words, if the prover forwards even a single bit of one of the responses, this is considered trivial help, as it gives the adversary an increased success probability in future rounds.
However, our definition allows such partial attacks as valid terrorist fraud attacks, concretely requiring that the prover's help does not give the adversary an *equal* advantage for future sessions, i.e. it only excludes attacks where the prover forwards the adversary any fragment of the *secret key itself*. We discuss the merits of both definitions in Section 2.3.7.
In the following, we show that this protocol does *not* attain terrorist fraud resistance in the sense of our definition above (see Section 2.1).

Theorem 2.19 (Reid et al. Properties) *Let* DB *be the distance-bounding authentication scheme in Figure 15 with parameters* $(t_{\max}, N_c)$*. This scheme has the following properties:*

- *It is neither impersonation resistant, nor terrorist fraud resistant (assuming the pseudorandomness of* PRF*).*
- *If the symmetric encryption scheme in this protocol is instantiated as bitwise XOR, this scheme is* not *mafia fraud resistant.*
- *If the symmetric encryption scheme used in this protocol is instantiated as bitwise XOR, and furthermore if the secret key sk is generated honestly at random from a distribution computationally indistinguishable from the uniform random distribution, this scheme is distance-fraud resistant; however, we cannot generalize this proof for arbitrary choices of* $\mathcal{E}$*.*

Proof. This protocol inherits the lack of impersonation resistance from the protocol due to Hancke and Kuhn [42] and so we do not show the proof here.
We first look at the distance-fraud resistance of this protocol (i.e. the second statement). The difference for distance-fraud resistance (with respect to the Hancke and Kuhn protocol) is that the responses computed here by the tag are eph and c, only one of which is computed by means of a PRF. Thus, the attack of Boureanu et al. [11] does not transfer trivially. However, the tag can, in a similar fashion, choose a convenient nonce that outputs a "weak" eph, such that, when it is later used as a key, it yields an output that has little entropy. This is possible since the IND-CPA notion applies to keys selected at random by a key-generation algorithm.
However, in case the symmetric encryption function $\mathcal{E}$ is instantiated as bitwise XOR, the Hamming distance between the two responses, eph and c is exactly *sk* at each execution. The proof goes as follows. First we replace *sk* by a uniform random value (and lose a term equalling the distinguishing advantage between the uniform random distribution and the distribution that *sk* is chosen from). Now we consider each round i in a distance-fraud attack. For this phase it holds

with probability $\frac{1}{2}$ that the bit c_i is different from the corresponding bit of $\text{eph}_i = c_i \oplus sk$. If $c_i = \text{eph}_i$ (this happens with probability $\frac{1}{2}$), the adversary forwards c_i and wins the round; else, if the two values are unequal, then the adversary has to guess which value to send (essentially predicting the challenge) and is successful with probability $\frac{1}{2}$.
However, if the symmetric encryption function *is* instantiated as a one-time-pad XOR operation, i.e. $c \leftarrow \text{eph} \oplus sk$, we cannot prove the protocol mafia fraud secure. A mafia fraud adversary could run the following attack, trying to recover the key *sk*. We will denote the adversary's guess of *sk* by *sk*′. The adversary begins by initiating a reader-adversary session sid_1 and an adversary-tag session sid_1^*. It relays all the lazy and time-critical rounds between the reader and the tag up to the $n-1$-th round (this is possible since the definition of mafia fraud resistance only excludes time-critical relaying —up to $T_{\max}$ rounds— for the session sid where the adversary succeeds in its authentication attempt). Finally, in the last round, it receives a challenge bit $b = R_{N_c} \in \{0,1\}$ in session sid_1. Then the adversary forwards the challenge $\bar{b} = b \oplus 1$ in session sid_1^*, receiving the value $T^{\bar{b}}$. The attacker finally forwards this value to the reader in session sid and waits to see if it is authenticated by the reader; if so, it sets $sk'_{N_c} = 0$, and else it sets $sk'_{N_c} = 1$.
There are two cases:

$b = 0$. Then, in sid_1, the reader expects c_{N_c} as a response. From session sid_1^*, where the adversary has forwarded $\bar{b} = 1$, the adversary has learned $\text{eph}_{N_c} = c_{N_c} \oplus sk_{N_c}$, which it forwards to the verifier; if the verifier accepts, then $c_{N_c} = c_{N_c} \oplus sk_{N_c}$, and thus $sk_{N_c} = 0$; in this case, the adversary's guess is correct, and $sk'_{N_c} = sk_{N_c}$. Else, if the verifier rejects, then $sk_{N_c} = 1$, and the adversary has again guessed correctly.

$b = 1$. In this case, the reader expects eph_{N_c}; the value forwarded by the adversary is $c_{N_c} = \text{eph}_{N_c} \oplus sk_{N_c}$. The same reasoning applies as above.

The adversary continues the attack in the same way, recovering the secret key bit-by-bit. Once the adversary has the complete $sk' = sk$, the adversary finally initiates its "challenge" session sid with the verifier, and a parallel session sid^* with the prover. The adversary forwards the lazy phase communication, and then queries the prover in advance to learn eph (i.e. it sends a challenge bit $R_i = 1$ for every round $i \in 1, \ldots, N_c$ in sid^*). Since these queries are made before the adversary receives the challenge in session sid, the adversary has not tainted the round. For every round in sid, if the verifier sends a challenge $R_i = 1$, then the adversary sends eph_i; else, if the verifier sends $R_i = 0$, then the adversary forwards $\text{eph}_i \oplus sk'_i$, and thus also responds correctly. Now the adversary authenticates with probability 1.
However, this attack is not extendable to arbitrary symmetric encryption schemes. What is needed in order for the scheme to achieve some mafia fraud resistance (equal to the mafia fraud resistance of the protocol of Hancke and Kuhn [42]) is for the encryption scheme to not leak any information about the key from learning both possible time-critical round responses for a give round. We elaborate more on such key-learning attacks and their consequences in Chapter 4.
Finally we show a terrorist adversary $\mathcal{A}$ for which there exists no simulator such that $\mathbf{Adv}^{\text{terror}}_{\text{DB}}(\mathcal{A}, \mathcal{S}, \mathcal{P}) \leq 0$. This would therefore show that the scheme is not terrorist fraud resistant. The idea is for the malicious tag $\mathcal{P}'$ to give information that facilitates the adversary's attack, without revealing any essential information about future impersonation attempts. Indeed, let $\mathcal{A}$ receive the value eph from $\mathcal{P}'$ in each of its $q_{\mathcal{V}}$ impersonation attempts. In the subsequent time-critical phases, if the reader sends challenge $R_i = 1$, $\mathcal{A}$ sends $T_i = \text{eph}_i$; else, the adversary guesses T_i. This adversary's probability to win is thus $\frac{3}{4}^{N_c} + \mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{E}}(\mathcal{A}'')$ for the adversary $\mathcal{A}''$ whose advantage to win against the IND-CPA of $\mathcal{E}$ is the largest.
Now consider a simulator $\mathcal{S}$. The simulator has no access to the tag $\mathcal{P}$, but it may run $\mathcal{A}$ internally. However, under the assumption of the pseudorandomness of PRF, there is only a negligible probability that $\mathcal{A}$ knows eph for any of the $q_{\mathcal{V}}$ impersonation sessions where the simulator attempts to authenticate to the reader. Thus, the simulator's probability of winning is $\frac{1}{2}^{N_c} + \mathbf{Adv}^{\text{IND-CPA}}_{\mathcal{E}}(\mathcal{A}'')$. Thus, the adversary has an advantage over any simulator $\mathcal{S}$. □

2.3.6 The Swiss-Knife RFID Distance Bounding Protocol

The Swiss-Knife protocol due to Kim et al. [50] aims to achieve privacy as well as mafia, terrorist, and distance fraud resistance. Very notably, the lazy phase of this protocol is divided into two parts: the first precedes the time-critical phase, the second follows it. In the first part, the reader and tag exchange nonces and the tag computes a pseudo-random function (PRF) on input a system constant and a tag-chosen nonce $N_{\mathcal{P}}$. The output of this function, a, is then XORed with the long-term secret key *sk*. In the time-critical rounds, the tag responds with either a or with $a \oplus sk$, depending on the reader's challenge. Note that if this protocol is run in an RFID scenario, the size of *sk*, which equals the number of time-critical rounds, is restricted by the tag's capacity to sustain time-critical rounds. Therefore, the keys are short. Finally, after the time-critical rounds, during the second lazy phase, the tag authenticates by computing the PRF on all the received challenges, its identity, and both the reader and the tag's nonces. The reader may then also authenticate by computing the PRF on input the tag's nonce. This second lazy phase is essential in preventing the recovery of the secret key in a key-learning attack, as in the protocol of Reid et al. In fact, the fact that the tag computes the second PRF value brings the mafia fraud resistance to loosely $(\frac{1}{2})$ per round.

In the Swiss-Knife scenario, each tag is associated with an identity $\mathcal{ID}$ which is stored by the reader in the same database that stores the secret key *sk* of the tag. In order to achieve anonymity, this identity is never sent, and the reader needs to search the database exhaustively to find it. This protocol also has some fault tolerance, i.e. the reader counts a total number of errors consisting of: (1) the number of faulty challenges R_i that the tag receives; (2) the number of faulty responses T_i that the reader receives; and (3) the number of rounds in which the tag's response exceeds the time threshold $t_{\max}$. The protocol is depicted in Figure 16. Note that Kim et al. also present a more efficient version, but whereas this second scheme is computationally more efficient than the simplified one, the security properties are comparable. In Figure 16, the value const is a system constant.

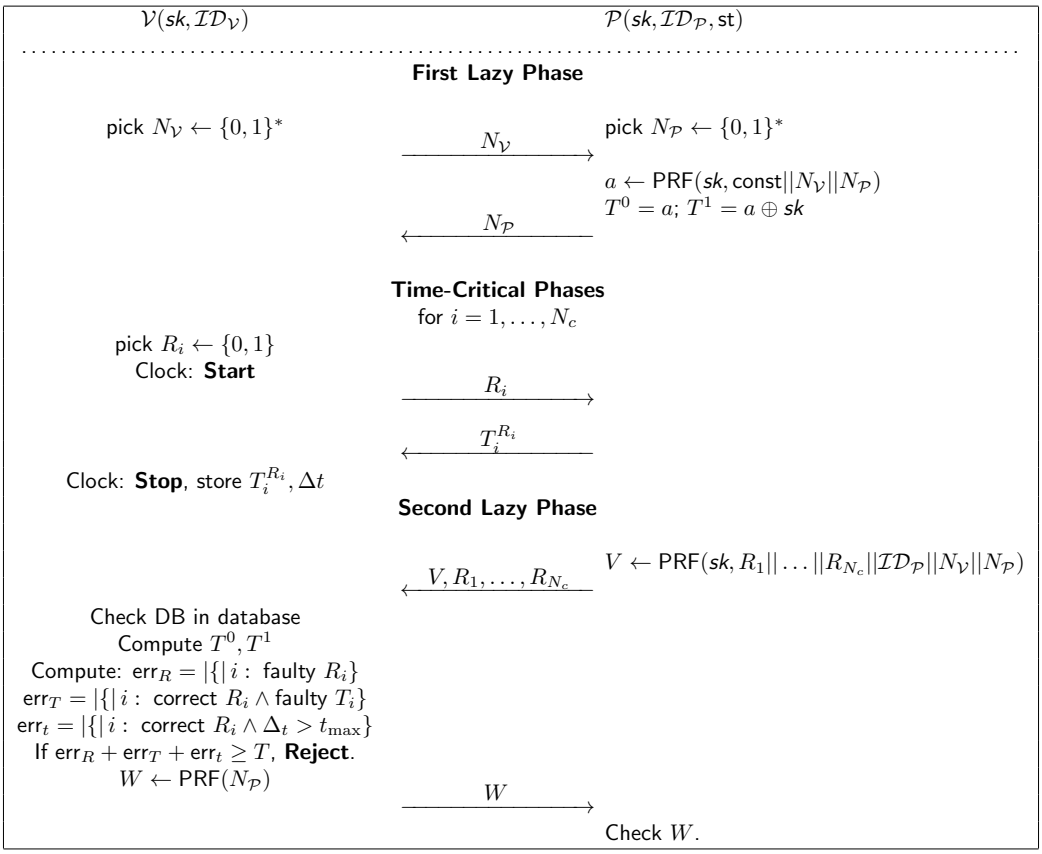

Figure 16: The Swiss-Knife protocol

Theorem 2.20 (Swiss-Knife Properties) *Let* DB *be the distance-bounding authentication scheme in Figure 16 with parameters $(t_{\max}, N_c)$, and assume the key sk is pseudorandom and chosen honestly, at random, before tag initialisation (in particular, the tag does not choose its own key), from a distribution $\mathcal{D}$ which is computationally indistinguishable from the uniform random distribution. This scheme has the following properties:*

- *It is not terrorist fraud resistant (assuming the pseudorandomness of* PRF*).*
- *For any $(t, q_\mathcal{V}, q_\mathcal{P}, q_{\text{OBS}})$-distance-fraud adversary $\mathcal{A}$ there exists a distinguisher $\mathcal{A}'$ against the distribution $\mathcal{D}$ such that:*

$$\mathbf{Adv}^{dist}_{\mathsf{DB}}(\mathcal{A}) \leq q_\mathcal{V} \cdot \left(\tfrac{3}{4}\right)^{N_c - T} + \mathbf{Adv}^{d}_{\mathcal{D}}(\mathcal{A}').$$

- *For any $(t, q_\mathcal{V}, q_\mathcal{P}, q_{\text{OBS}})$-mafia-fraud adversary $\mathcal{A}$ against the scheme there exists a (t', q')-distinguisher $\mathcal{A}'$ against*

PRF *(where $t' = t + O(n)$ and $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$) such that*

$$\begin{aligned}\mathbf{Adv}^{\textit{mafia}}_{\text{DB}}(\mathcal{A}) \quad \leq \quad & \left(\tfrac{1}{2}\right)^{N_c - T} + 2\binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-(|N_{\mathcal{V}}| + \lceil \frac{N_c}{2} \rceil - T)} + 2\binom{q_{\mathcal{P}}}{2} \cdot 2^{-(|N_{\mathcal{P}}| + \lceil \frac{N_c}{2} \rceil - T)} \\ & + q_{\mathcal{V}} \cdot \mathbf{Adv}^{d}_{\text{PRF}}(\mathcal{A}') + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-(|N_{\mathcal{V}}| + N_c - 1 - T)} + 2\binom{q_{\mathcal{P}}}{2} \cdot 2^{-(|N_{\mathcal{P}}| + N_c - 1 - T)}.\end{aligned}$$

- *For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-impersonation adversary $\mathcal{A}$ against the scheme there exists a (t', q')-distinguisher $\mathcal{A}'$ against* PRF *(where $t' = t + O(n)$ and $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$) such that*

$$\mathbf{Adv}^{\textit{imp}}_{\text{DB}}(\mathcal{A}) \quad \leq \quad q_{\mathcal{V}} \cdot 2^{-|V|} + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-(|N_{\mathcal{V}}| + N_c - T)} + \mathbf{Adv}^{d}_{\text{PRF}}(\mathcal{A}') + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-(|N_{\mathcal{P}}| + N_c - T)}.$$

Proof. The proof of statement 1 is trivial: the malicious tag can even send the adversary the secret key *sk* and the output V for the second lazy phase. However, the simulator cannot guess the value of $\mathcal{ID}_{\mathcal{P}}$ except with probability $2^{|\mathcal{ID}_{\mathcal{P}}|}$.
For the second statement, note that the tag does *not* choose the key *sk*, thus, if this key is chosen at random from a distribution computationally indistinguishable from the uniform random distribution, the attack of Boureanu et al. [11] is thwarted. Indeed, with great probability, whatever the value a output by the PRF for this session, the value $a \oplus$ *sk* is with high probability at a large Hamming distance from a. The proof goes as follows. First we replace *sk* by a uniform random value (and lose a term $\mathbf{Adv}^{\text{d}}_{\mathcal{D}}(\mathcal{A}')$). Now we consider each round i in a distance-fraud attack. For this phase it holds with probability $\frac{1}{2}$ that the bit $T_i^0 = a_i$ is different from the corresponding bit of $T_i^1 = a_i \oplus$ *sk*. If $T_i^0 = T_i^1$ (this happens with probability $\frac{1}{2}$), the adversary forwards T_i^0 and wins the round; else, if the two values are unequal, then the adversary has to guess which value to send (essentially predicting the challenge) and is successful with probability $\frac{1}{2}$. After we account for the fault tolerance level T we attain the above-stated bound.
For the third statement, the proof goes slightly differently than for previous protocols, e.g. the scheme due to Kim and Avoine 2.3.4. In particular, the response strings T^0 and T^1 are now related. The proof follows in these rough steps: (1) assuming that the secret key *sk* is indistinguishable from a random string of appropriate length *at the end of any mafia fraud interaction*, we can prove mafia fraud security as in the previous proofs, in particular replacing the response strings by random values. We also account, even for sessions with matching $N_{\mathcal{V}}$ and $N_{\mathcal{P}}$, for about half the challenges in time-critical rounds. Then, we show that (2) except with negligible probability the adversary cannot distinguish the key *sk* from a random string of corresponding length. This second step is proved as follows: note first that if the adversary merely observes the interaction between an honest prover and an honest verifier, or simply relays all messages exactly as he receives them, this does not, with great probability, reveal any information about *sk*. However, if two sessions share the same nonces (and the computed responses are identical), the prover may learn about $\lceil \frac{N_c}{2} \rceil$ of the bits of the secret key (wherever the challenges differ and the prover gives the correct response from the other response string). This happens with probability roughly

$$\binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-(|N_{\mathcal{V}}| + \lceil \frac{N_c}{2} \rceil - T)} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-(|N_{\mathcal{P}}| + \lceil \frac{N_c}{2} \rceil - T)}.$$

We now assume this is not the case. Now for each of the N_c time-critical phases the adversary learns a bit from either one of the two response strings, but not from both, thus leaking no information about the secret key. Furthermore, if the adversary *does* interfere with the running of the protocol, in particular changing either a challenge or a response, there are only three possibilities: (a) the adversary changes at least one challenge or one response, in which case the matching prover computes a different authentication value in the matching adversary-prover session than the value expected by the verifier in the verifier-adversary session, and the adversary is unable to provide the correct authentication value (in this case the adversary can learn nothing beyond what it learns in a common observation of an authentication attempt); (b) the adversary has seen (or "created", by forwarding nonces) another session in which all the challenges and responses are exactly the same for all except the rounds where the adversary wishes to change the challenge and/or response (thus the adversary has seen a valid authentication string V for the altered string of challenges and responses); (c) the adversary is able to come up with a forgery for the value V. Event (b) happens with probability:

$$\binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-(|N_{\mathcal{V}}| + N_c - 1 - T)} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-(|N_{\mathcal{P}}| + N_c - 1 - T)}.$$

Event (c) happens with probability $\mathbf{Adv}^{\text{d}}_{\text{PRF}}(\mathcal{A}')$ (per verifier-adversary session). This accounts for the bound above.

The proof for impersonation security runs more or less as in the previous proof, only now we only account for the probability of lazy-phase authentication. □

2.3.7 The Case for Terrorist Fraud Resistance

In this section, we prove that two constructions which claim to achieve terrorist fraud resistance are in fact *not* terrorist fraud resistant in the framework we present in Section 2.1. Additionally, it appears that intuitive countermeasures against terrorist fraud resistance do not work, as *partial* help from the dishonest prover gives the adversary some advantage over the simulator.
Our results may be viewed from two separate points of view. It can be argued, on the one hand, that our model due to Dürholz et al. is too strong, and does not accurately capture the notion of terrorist fraud resistance. On the other hand, our results may be viewed as proof that terrorist fraud resistance is in fact a very powerful attack, which is difficult to counteract in practice. We present and assess both points of view in the considerations below.

Model strength. As noted in Section 2.3.5, the notion achieved by [70] is very weak in the sense that it excludes even prover information that significantly aids adversary authentication while disclosing a relatively insignificant *part* of the secret key. We note that previous definitions, such as the one in Avoine et al.'s framework [4], are ambiguous regarding this point (see Section 2.2.2). In fact, Avoine et al. require, literally, that the prover's help gives the adversary no advantage in future attempts. It is unclear, however, what "further" means in this context: does it refer to the success probability of the adversary *after* the prover helped it, compared to the adversary's success *before* the prover helped it, or rather to the notion we capture here, i.e. the success probability of the adversary *after* the prover helped it compared to *while* the prover helped it? If the definition of Avoine et al. is to be taken in the sense outlined by [5], it appears that the help forwarded by the prover must preserve the secrecy of the secret key in an information-theoretical way. This notion seems too strong, as it should be possible to learn some information about the key as long as this information is not *useful* to an adversary (simulator) after the prover has withdrawn its support.
If we take the former, weaker notion, the protocol due to Reid et al. [70] is intuitively terrorist fraud resistant. However, we point out that the only concurrent framework covering distance-bounding, namely the framework of Avoine et al. [4] does *not* precisely formalise the weaker definition of terrorist fraud resistance above. Recognising that a more relaxed game-based definition of terrorist fraud resistance may yet be useful, however, we also consider in Chapter 5 a formalisation of the notion sketched above and prove that this can be achieved by existing protocols.
A further, quite separate question is which definition best captures the intuition behind terrorist fraud attacks. A strong degree of terrorist fraud resistance is always more desirable, thus from this point of view our definition sets the standard for protocol design. On the other hand, this definition seems hard to achieve, as it enables attacks where some indirect information about the key is forwarded to the adversary (as in Sections 2.3.5 and 2.3.6).
The intuition of terrorist fraud resistance is that the malicious prover is willing to assist the adversary in its authentication attempt, but wants to control his access. Thus, the adversary should not be able to authenticate without the prover's help. We note, however, that the adversary always has some (usually negligible in the number of time-critical rounds) probability of authenticating without the prover's help: this is equivalent to the probability that he guesses the correct replies or, equivalently, that he guesses the secret key.
How far does our model cover this intuitive notion? We quantify the adversary's success probability in the presence of the malicious prover, and then the simulator's probability (where the simulator does not have access to the prover, only somewhat to the adversary in the state when communicating with the tag). The scheme is considered terrorist fraud resistant if the simulator's probability of success equals (or is greater than) the adversary's probability of success. In other words, an attack is successful if the prover's help enables the adversary to succeed in one session with some probability, but this probability diminishes in future sessions, when the prover is no longer available. In this scenario the prover has the guarantee that the adversary will only be able to authenticate (afterwards) with less probability. This definition also seems too strong, in the sense that we accept an attack where the prover authenticates with probability 75% (3 out of 4 times), but the simulator can only authenticate with probability 50% (1 out of 2 times). This contradicts the spirit of terrorist fraud resistance as it is understood in the literature.
A middle way would be to define a so-called tolerance level for the simulator, i.e. accept attacks as long as the simulator's success probability does not exceed this tolerance level. Note, however, that the attack presented in Section 2.3.5 can be tweaked so that the adversary still has an advantage over the simulator, whereas the simulator succeeds with a probability within the tolerance level (instead of giving half the response, the prover would forward only a number of bits of this response, thus easing the adversary's job).
It is our opinion that our notion, though strong, does capture the intuition of terrorist fraud resistance better than the weaker definition which these protocol seem to attain. A common approach in security is to be conservative and to ask

for strong(er) security, rather than to label insecure protocols as secure.

Constructive aspects. A second perspective in which to view our result is a constructive one, i.e. if we consider that our model captures the correct notion of terrorist fraud resistance, then clearly achieving this definition requires a stronger construction. One might argue that the strong requirement posed by our model would lead to inefficient constructions. We argue, however, that the notion of terrorist fraud resistance, is, in its own right, a very strong notion: here, the (dishonest) prover *helps* the adversary authenticate. The challenge is thus to ensure that *any* information leaked to the adversary automatically will carry over to the simulator. We in fact show a construction that achieves terrorist fraud resistance in the chapter especially dedicated to this attack, i.e. Chapter 5.
We also note that there is a clear separation between distance-bounding realisations for RFID and for more powerful devices. Indeed, terrorist fraud resistance might be more easily achieved if it is possible to use, say, public key cryptography. In this sense, we could wonder how realistic a threat terrorist fraud attacks are on RFID systems and whether it is worth addressing them directly in protocol design. With RFID tags used in the pharmaceutical industry, in general logistics, and in public transport [21, 77], it seems quite likely that terrorist fraud attacks are quite likely in practice in these settings. In fact, RFID systems are also used in airport security in many German airports: impersonation MITM attacks have already been mounted on these systems by the Chaos Computer Club (CCC) [37]. Though these attacks were not real-time relay attacks, the incentive to mount mafia and terrorist fraud attacks on RFID authentication protocols is rather high. It remains an open question whether RFID systems can be efficiently protected against terrorist fraud in practice, however. The results in this paper show that terrorist fraud resistance is not trivial to achieve, and that achieving it may be inefficient for RFID devices. As terrorist fraud resistance is, however, both a very strong, and a very desirable goal, we interpret our results in this section as an incentive to construct protocols that *are*, in fact, terrorist fraud resistant in our strong definition.

2.3.8 Conclusions: Protocol Comparison

We finally recapture the tools used by various constructions to attain distance-bounding security goals, particularly for RFID scenarios, and look again especially at the hardness of achieving terrorist fraud resistance.
Clearly, the mafia fraud resistance of a protocol depends on the success rate of the Go-Early strategy: in particular, the adversary should not be able to use information leaked in advance from the prover. The adversary's success rate is high for the Hancke and Kuhn protocol [42] (which is vulnerable to Go-Early attacks). To address this, the protocol due to Kim and Avoine [49], whose security is assessed in [27] and in Section 2.3.4 above, is to give some form of reader authentication during the time-critical rounds: thus, the honest tag has a higher chance of detecting an adversary leeching its responses. Creating a dependency between the responses as in [6] is also a solution, but this protocol involves exponential storage. Finally, the solution due to Kim et al. [50] seems quite elegant: by authenticating the challenges after they are sent, the tag makes sure that even if the adversary leeches some challenges, it can only receive the correct authentication response if it has guessed *all* the challenges in advance. This latter solution, however, involves a second PRF computation.
As outlined in the previous section, the literature is unable to provide any effective means to achieving terrorist-fraud resistance. Creating an interdependency between the secret key and the tag responses has thus far been the advertised strategy towards attaining terrorist fraud resistance. However, the attack we show in this paper against Reid et al.'s protocol [70] can be used against *any* scheme. We show in Chapter 5 that terrorist fraud resistance *can* in fact be achieved even in the strong notion presented in Section 2.1.2. We also show that the construction due to Reid et al. attains a weaker form of terrorist fraud resistance, which we also formally define in Chapter 5.
Finally, we note that the attack of Boureanu et al. [11] affects most distance-bounding protocols in the literature, as they all rely on an incorrect assumption regarding the pseudorandomness of the underlying primitives used. One solution to fix this problem is, as mentioned in the introductory paragraphs of this chapter, to add a verification step, where the reader will reject a prover who forwards a "weak" nonce, while the prover will have to generate nonces until some minimal entropy is achieved in the response. However, this approach has the drawback of adding communication and computation complexity to the protocol (as the prover might be forced to select multiple nonces and every time compute the corresponding PRF value). Another approach is that of the Swiss-knife protocol, which relates the two outputs of the pseudorandom function by means of the secret key. However, by relating the two responses, the protocol becomes now vulnerable to key-learning attacks as defined in Section 4.

3 Static Keys vs. Key Update

In the previous chapter, the prover and verifier, resp. reader and tag, shared a static key, i.e. a key which is never updated. However, a rising concern in the context of authentication and distance bounding in general is privacy. In the context of RFID authentication, privacy is defined as follows: an adversary must not distinguish which valid tag interacts with the reader. An early RFID privacy model was introduced by Juels and Weis [47]; their notion was based on tag indistinguishability (an adversary cannot tell which of two valid tags is authenticated). In 2007, Vaudenay [73] formalised an RFID security and privacy model where adversaries can corrupt tags and learn secret keys; this model was later refined by Paise and Vaudenay [64], and Ng et al. [59]. Vaudenay introduced eight types of adversaries, where narrow-weak adversaries (see below) coincide somewhat with Juels and Weis' adversaries, as they do not corrupt tags. A fundamental goal in authentication, however, remains to achieve better (stronger) privacy. Following a recent analysis by Armknecht et al. Armknecht2010, who noted some issues regarding the blinder in the model due to Vaudenay [73], Canard et al. Canard2010 also proposed a simulation based model that resolves these issues by introducing a trivial adversary. A recent game-based notion due to Hermans et al. [43] captures the same adversary classes in the work of Vaudenay, but from a game-based, indistinguishability perspective. In this work, we choose the model due to Vaudenay [73], as it is the most commonly used model for RFID privacy, and since the issues pointed out by [2] do not affect our proofs.
The adversary classes defined by Vaudenay [73] are: weak adversaries (who do not corrupt tags); forward adversaries (once these adversaries corrupt a tag, they can only continue the attack by further corruption queries); destructive adversaries (who destroy tags once they corrupt them); and strong adversaries (who are free to choose their strategy as they wish, with no restrictions). Furthermore, an adversary in any of these classes can also be *narrow* (if the adversary does not learn the result of the authentication attempt) or *wide* otherwise. Vaudenay showed that his notion of strong privacy cannot be achieved. Thus, if the adversary can corrupt tags at any point *and* learn the output of authentication sessions, it always breaks privacy. *Narrow* strong privacy, i.e. when the adversary does *not* learn authentication output, requires key agreement, which in turn requires public key cryptography, a primitive deemed too expensive for most RFID tags[9]. However, Vaudenay shows how to achieve so-called *narrow-destructive* privacy, where the adversary destroys the tags upon corruption (as is the case when the adversary damages tags to learn their secret keys): the idea is to use key updates, thus ensuring that corruption only reveals an ephemeral secret, and no further information about the past states of the key.

Contributions. In this chapter we address security in distance-bounding protocols in the context of key updates. The main achievements we show here are found in [60]. Concretely we show a *formal model* capturing the notion of distance bounding in the setting of key updates. Towards this goal we define long-term completeness (i.e. availability) for distance bounding. Also, we show a compiler that turns any mafia fraud, distance fraud, and impersonation resistant, narrow-*weak* private distance-bounding RFID protocol into a narrow-*destructive* private distance-bounding RFID protocol with the same distance-bounding properties, and with long-term completeness. Our construction also requires that the key-generation algorithm of the underlying distance-bounding protocol outputs pseudorandom keys K[10]. Concretely, we wrap a construction like Vaudenay's narrow-destructive protocol [73] around an underlying distance-bounding scheme, such that the reader and tag both update state by using a pseudorandom function (PRF). The prover updates state early, after an initial lazy authentication-phase (this prevents the adversary from desynchronising the reader from the tag by simply dropping messages, since with high probability it is unable to authenticate afterwards). Contrary to Vaudenay's construction [73], the reader *also* updates state, but only if the tag succeeds in (a) an initial authentication phase; (b) the distance-bounding authentication steps. Finally, though the adversary can make the tag update state more often than the reader does, the reader is then able to "catch up" with the tag's state in an initial state-recognition step. Thus, we prevent denial-of-service (DoS) attacks and gain *availability* and efficiency in the reader's computations for the compiled protocol (compared to the initial scheme in [73]).
Concretely, we:

- Formalise availability (DoS resistance);
- Describe a compiler where the reader and tag update state at each successful authentication. Though the adversary may force the tag to update before the reader, the reader can "catch up" at the next honest authentication attempt. This is an efficiency improvement with respect to the construction in [73], where the reader must always catch up from scratch with the original key. This compiler will be of great use in achieving two stronger flavours of mafia fraud resistance, which are presented in Chapter 4.

[9] Notably, more expensive tags *do* enable elliptic curve cryptography [52, 53, 75]; however, passive and semi-active tags generally cannot run public key cryptography. An interesting direction for future research would be indeed to investigate better privacy levels for tags that can run public key cryptography.

[10] This is not a very strong assumption, as such algorithms are usually required to produce pseudorandom outputs.

- Prove that our compiler preserves mafia, distance, and impersonation security in the sense of the previous chapter. We also discuss why our compiler does *not* provably preserve terrorist fraud resistance. Namely, it is hard to formalise how a simulator may use partial update-information that the adversary receives from the tag. Note also that many distance-bounding protocols do not address this attack [6, 13, 42, 49]; also, no protocol in the literature *claiming* to achieve terrorist fraud resistance is, in fact, *provably* terrorist fraud resistant in our strong definition. We also note that, while for e.g. the protocol we show in Chapter 5, which *is* terrorist fraud resistant, it is hard to construct a simulator as required by our definition in the previous chapter, it is also not immediately clear how one would construct a concrete terrorist fraud attack against a compiled version of this protocol.
- Discuss optimisations of our compiler for two particular cases, namely for distance-bounding protocols which do not use reader authentication, and for protocols with reader authentication and an initial lazy phase achieving impersonation security by a pseudorandom function (PRF).

Related work. As the most important related work, we review the framework of Vaudenay [73] in Section 3.1. In this paragraph we briefly mention other related work.
Different simulation-based privacy frameworks were introduced in [23, 56], the latest one describing a very strong Zero-knowledge based notion of privacy. Very recently, a game-based security and privacy model was introduced by Hermans et al. [43]. However, simulation-based privacy is, arguably, too strong, capturing the notion that the adversary should not only be unable to distinguish a legitimate tag (with or without corruption), but they should, in fact, be unable to tell anything about this tag (including whether it is legitimate or not). Whereas such strong privacy is desirable, achieving destructive privacy in the framework of Vaudenay [73] is an important first step in achieving privacy for distance-bounding protocols.
Privacy can also be achieved as shown in 2009 by Sadeghi et al. [71], i.e. by means of anonymizers, which are corruptible third parties in the setting of [73]. As anonymizers are independent —and not necessarily trusted— parties in RFID networks, the security model needs to include the communication between tags and anonymizers. The construction proposed by Sadeghi et al. [71] includes three actors: readers, tags, and anonymizers, and two protocols: anonymization and identification. Thus, a parallel approach to ours could use anonymization instead of internal key updates in a similar way as we describe here.
Finally, Cremers et al. [22] introduced a further attack against distance-bounding protocols, i.e. distance hijacking attacks, where a malicious tag commits distance fraud in the presence of an honest tag. As our paper focuses on privacy, we do not address this attack, but we stress that a formalised model for the single-reader multiple-tags scenario is highly desirable, and in such a setting, hijacking attacks are essential.

3.1 Preliminaries

In this section, we briefly review the privacy model due to Vaudenay [73]. Then we extend the security framework from the previous chapter to provide for dynamic keys. In this stateful context, we give a definition of long-term completeness, a notion we call *availability*.

3.1.1 Review of Privacy Model

We briefly review the privacy framework of Vaudenay [73], who considers RFID systems consisting of a single reader, but multiple tags. For the privacy game, tags are associated with handles called *virtual tags* (vtags). The adversary can send messages to the reader and to virtual tags, it can "draw" and "free" vtags (thus assigning handles to tags), it can observe honest reader-tag interactions, and it can also corrupt tags (learning their state, including the secret key). Vaudenay models the adversary's behaviour by means of a number of oracles that $\mathcal{A}$ can access, and which enable its interaction with the system. We refer to the original paper [73] for more details about the precise syntax of the model.
The four major adversary classes in [73] differ in the way an adversary acts upon corrupting tags (see below). Furthermore, adversaries are *narrow* if they don't know the protocol output (i.e. whether a tag has been accepted or not) and *wide* otherwise. Narrowness is an additional property, which can be combined with any of the following four adversary classes:

Weak adversaries,. who cannot corrupt tags.

Forward adversaries,. who, once they have used the tag corruption oracle, may only use this oracle (in particular, the tag cannot run protocol steps or full executions).

Destructive adversaries,. who destroy the handle of the current tag upon corrupting it. Note that they are still allowed to interact with the RFID system arbitrarily with respect to other tags.

Strong adversaries,. who may use all the articles arbitrarily.

The adversary's success is measured with respect a simulator $\mathcal{B}$ called a Blinder, which simulates all queries except corruption queries to the blinded adversary $\mathcal{A}^{\mathcal{B}}$. The adversary's goal is to distinguish between the real RFID system and an interaction with the blinder, after playing a two-phase game. In the attack phase, $\mathcal{A}$ interacts with all oracles arbitrarily, subject to corruption query restrictions; then, in the analysis phase, the adversary does not access any of the oracles, but receives the secret table containing the correspondence between tag identities and the respective handles. Finally, the adversary returns a bit denoting its success ($b = 1$) or failure in the attack ($b = 0$). Write $\mathcal{A}$ for the adversary, and $\mathcal{A}^{\mathcal{B}}$ for the blinded adversary, and denote:

$$\mathbf{Adv}^{\mathsf{priv}}_{\mathsf{DB}}(\mathcal{A}) = \mathrm{Prob}[\,\mathcal{A}\text{ wins}] - \mathrm{Prob}\Big[\,\mathcal{A}^{\mathcal{B}}\text{ wins}\Big]$$

Privacy is then defined as follows [73].

Definition 3.1 (Privacy) *Let* DB *be a distance-bounding authentication scheme with timing parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$. *Let* $\mathcal{P}$ *denote one of the adversary classes defined above. The scheme* DB *is* $\mathcal{P}$*-private if for any adversary* $\mathcal{A}$ *there exists a blinder* $\mathcal{B}$ *such that* $\mathbf{Adv}^{\mathsf{priv}}_{\mathsf{DB}}(\mathcal{A})$ *is negligible.*

We note that the privacy of a protocol depends on what information is leaked by corrupting tags. Paise and Vaudenay [64] showed that narrow-forward privacy cannot be attained if the entire state of the tag leaks. In this paper, we assume that corrupting the tag yields a part of the state consisting of two secrets, which we denote S and S^*, where $S^* = F^i_{sk_F}(S)$ for a given pseudorandom function F, a long-term secret sk_F, stored apart from S and S^*, and a positive integer i. However, we assume that corruption does not leak two further long-term secrets, sk_F and sk_G (the keys to two pseudo-random functions F and G); furthermore, we assume that intermediate computations and the implementations of F and G, as well as those of an additional pseudorandom function, which we define as PRF, are not leaked. We note that we can assume that the secret keys sk_F and sk_G are hard-coded into the pseudorandom functions F and G. This is possible, since neither sk_F, nor sk_G need be tag-specific, and they are never updated.

3.1.2 Availability

In the context of key updates, the notion of completeness —i.e. the property that a reader always accepts a legitimate tag— is no longer static: the adversary can cause a desynchronisation between reader and tag states, such that a legitimate tag is unable to authenticate. This attack is different from simple jamming attacks, which also result in legitimate provers being unable to authenticate: in particular, jamming attacks (which are always possible) assume that the adversary is online all the time, so that it jams the communications. Furthermore, if a communications' area is jammed, it is possible to change the positions of the prover and verifier, such that they overcome the jamming. However, if the prover and verifier states are desynchronised, the adversary no longer needs to stay online, and re-positioning won't help: the prover will no longer be able to authenticate to the verifier *because the verifier no longer considers the prover's state to be legitimate.* Many authentication protocols featuring key updates are vulnerable to such denial-of-service (DoS) attacks, e.g. the two protocols based on YA-TRAP of Chatmon et al. [19]. Note that DoS attacks are one successful way of breaching privacy as defined in the previous section.
We define *availability* as long-term completeness. The adversary may interact arbitrarily with the tag and the reader, also relaying messages. In particular, adversaries may choose to drop messages from honest reader-tag communication. At one point, the adversary stops, and the tag and reader interact in a single round (the adversary is in observation mode). We say that the adversary wins if the (honest) reader outputs a 0 bit. In other words, the adversary wins if (by arbitrary interaction with the reader and the tag), it makes the honest tag unable to authenticate in an honest session with the reader.

Definition 3.2 (Availability) *Let* DB *be a distance-bounding authentication scheme with timing parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$. *A* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$ *adversary* $\mathcal{A}$ *wins against availability if the reader rejects in one of the* q_{OBS} *reader-tag sessions* sid. *We denote by* $\mathbf{Adv}^{\mathsf{avbl}}_{\mathsf{DB}}(\mathcal{A})$ *the probability of* $\mathcal{A}$ *winning.*

In the context of DoS attacks, a classical attack appears when the reader iteratively updates state for a finite maximum number of times t, in order to "catch up" with the tag. In this case, the adversary can break availability by making the tag update state $t + 1$ times, such that the reader never recognises it as valid again. However, if we modify the protocol such that the reader must simply state until it identifies the tag (i.e. we take t to be infinitely large), this enables a different type of attack, where the adversary simply sends nonsense to the reader, who will be caught in an endless verification loop. This attack is much more difficult to prevent without allowing an adversary to breach privacy. Our approach requires that an adversary is unable to leak part of the tag state, as well as the precise implementation of the pseudo-random functions required for the computation. We give more details in the following section.

3.2 Our Compiler

We proceed to describe a compiler preserving impersonation resistance, as well as mafia and distance fraud resistance, while attaining narrow destructive privacy and availability.

In particular, the compiler we present does *not* provably preserve terrorist fraud resistance. We discuss the difficulties of attaining this very strong property after we describe the construction, and we note that attempting to gain terrorist fraud resistance would come at greater computational cost and a compromise in security. In particular, the main difficulty here is achieving *provable* terrorist fraud resistance in a very strong sense (i.e. our simulation-based definition from Chapter 2). We also note that even the protocols in the literature which *attempt* to address terrorist fraud attacks are not provably secure: indeed, we have shown in Chapter 2 a generic attack against such protocols. Thus, the difficulty in attaining terrorist fraud resistance does not stretch only as far as our compiler, but also to generic distance-bounding protocols.

However, mafia and distance fraud resistance are the most significant attacks against distance-bounding protocols. Notably, the protocols outlined in the previous chapter, all aim to thwart such attacks (even though distance-fraud resistance seems harder to achieve than previously thought), whereas only a small minority aims to achieve terrorist fraud resistance. Thus we also aim, by our protocol, to preserve mafia and distance fraud resistance, together with the basic authentication requirement of impersonation resistance.

We consider a general distance-bounding protocol $(\mathsf{Kg}, \mathcal{V}, \mathcal{P})$ for parameters $(N_c, t_{\max}, T_{\max}, E_{\max})$ as outlined in the previous chapter. Here the reader and each legitimate tag share an initial key K generated by Kg (this key will then be updated; thus, it is possible for a reader and tag not to share the same key at the beginning of a protocol execution; the reader, however, will then have the possibility to update its state). In particular, the tag and reader keep internal states, which we denote $\mathsf{st}_{\mathcal{P}}$ and resp. $\mathsf{st}_{\mathcal{V}}$. These variables are both instantiated with copies of the original key K, i.e. $\mathsf{st}_{\mathcal{P}} = (K, K)$ and resp. $\mathsf{st}_{\mathcal{V}} = K$ at key generation. Both tag states are suspect to corruption attacks. We will use two additional keys sk_F and sk_G, stored by both reader and tag in a component that does not leak information during corruption; these keys are used for two pseudorandom functions F and G. Of these two functions, F is used for key updates, whereas G is used to transform the first part of the state during the protocol run, achieving unlinkability at the same time as availability.

As discussed in the introductory paragraphs of this chapter, there are two additional requirements for the distance-bounding protocol: (1) We require that the outputs of Kg are pseudorandom (i.e. indistinguishable from random), but make no assumption regarding the structure of this protocol; and (2) the protocol must be narrow-weak private in the sense of [73]. Our compiler can be used on such generic protocols to build new distance-bounding protocols $(\mathsf{Kg}^*, \mathcal{V}^*, \mathcal{P}^*)$ for parameters $(N_c, t_{\max}, T_{\max}, E_{\max})$ which are narrow destructive-private in the sense of Vaudenay [73]. Our compiler preserves (the exact levels of) mafia and distance fraud resistance, as well as impersonation security, and grants the new protocol availability (long term completeness).

3.2.1 Compiler Description

The compiler takes as input a distance-bounding protocol $(\mathsf{Kg}, \mathcal{V}, \mathcal{P})$ with the following properties: (1) $(t^{\mathsf{mafia}}, q_{\mathcal{V}}^{\mathsf{mafia}}, q_{\mathcal{P}}^{\mathsf{mafia}}, q_{\mathrm{OBS}}^{\mathsf{mafia}}, \epsilon^{\mathsf{mafia}})$ resistant to mafia fraud attacks; (2) $(t^{\mathsf{mafia}}, q_{\mathcal{V}}^{\mathsf{dist}}, \epsilon^{\mathsf{dist}})$ resistant to distance fraud attacks; (3) $(t^{\mathsf{imp}}, q_{\mathcal{V}}^{\mathsf{imp}}, q_{\mathcal{P}}^{\mathsf{imp}}, q_{\mathrm{OBS}}^{\mathsf{imp}}, \epsilon^{\mathsf{imp}})$ secure against impersonations; and (4) narrow-weak private in the sense of Vaudenay [73]. Additionally, we require that the values K output by the key generation algorithm Kg are pseudorandom. The compiler outputs a protocol $(\mathsf{Kg}^*, \mathcal{V}^*, \mathcal{P}^*)$ having the same properties $(1) - (3)$, as well as being (4^*) destructive-private in the sense of Vaudenay [73]. The latter property also implies availability.

Main Idea (informal). A naïve approach would be to simply use the narrow-destructive private protocol due to Vaudenay [73]. In this protocol, the reader and tag also use two pseudorandom functions, where one is used to compute an ephemeral secret, used for authentication, and the other function is used to update the key. The central idea is that the tag always updates state, but the reader never updates state and instead iteratively updates the key at verification in order to re-synchronize with the tag. We can improve the efficiency of the protocol by allowing the reader to also update state at the end of a successful authentication attempt.

However, this approach does not prevent denial of service (DoS) attacks where the adversary simply transmits nonsense to the reader, making it enter an endless search loop. Thus we enhance the protocol due to Vaudenay [73] such that the tag keeps an additional state, apart from the current iteration of the key. We denote the current key iteration by $\mathsf{st}_{\mathcal{P}}^1$; the additional state kept by the tag, denoted $\mathsf{st}_{\mathcal{P}}^0$, is the last iteration of the key for which the tag is convinced that the reader has updated state. On the reader side, the reader will only have to check a single iteration of the secret key; then, relying on the correctness of $\mathsf{st}_{\mathcal{P}}^0$, the reader will "catch up" with a legitimate tag. The main tricks we use are: (1) the tag updates (at least a part of the) state early, before the distance bounding is run; (2) the reader only updates state after

a successful authentication session; (3) in the initial phase of the protocol, the reader uses lazy phase authentication to "catch up" with a legitimate tag if it is convinced of the tag legitimacy: i.e. if the tag has updated state more often than the reader, the reader now catches up with a the current state of a legitimate tag; (4) though the first part of the state, $\mathsf{st}^0_{\mathcal{P}}$, is not updated at every authentication attempt, the pseudorandom function G ensures that the adversary cannot link authentication attempts based on this value.
These four strategies help us achieve availability as follows. Due to trick number (3), the adversary can only desynchronise the tag and reader if the tag does *not* update state, while the reader *does* (else, the reader catches up at the next session with the legitimate tag). Since we use trick number (1), desynchronisation occurs if the tag does *not* engage in the distance-bounding protocol. However, trick number (2) implies that the adversary needs to, on its own, authenticate to the reader, a fact prevented by the mafia *and* impersonation security of the underlying distance-bounding protocol. Finally, trick number (4) ensures that an adversary leaking the first part of the state cannot associate authentication sessions.
Apart from availability, an additional problem must be considered in the compiler design, due to the compiler's generality: in protocols like that of Kim and Avoine [49], the reader $\mathcal{V}$ generates its time-critical round input based on the secret key. Thus, in order to run an underlying, stateless protocol as a black box, the reader must know the tag's state before the protocol run. In our compiler, this is achieved by a round of PRF-based authentication before the protocol run. This step, however, is not necessary if the underlying protocol can be run (up to the verification steps) without knowledge of the secret key. We show some optimisations of the compiler after describing and discussing the general construction. Note also that the tag updates state just after this state recognition step.

Formal Description. Let $\mathsf{DB} = (\mathsf{Kg}, \mathcal{V}, \mathcal{P})$ be a distance-bounding authentication scheme. We describe the compiled scheme $\mathsf{DB}^* = (\mathsf{Kg}^*, \mathcal{V}^*, \mathcal{P}^*)$. The key generation algorithm Kg^* runs Kg as a black box, generating the pseudorandom key K. The tag and reader keep internal states instantiated with as $\mathsf{st}_{\mathcal{P}} = (K, K)$ and resp. $\mathsf{st}_{\mathcal{V}} = K$. Then Kg^* also generates the secret keys sk_F and sk_G, which are shared by the reader and tag. The algorithms $\mathcal{V}^*$ and $\mathcal{P}^*$ are changed as depicted in Figure 17. Let F, G, and PRF denote pseudorandom functions (PRFs) independent of any other PRFs used by the distance-bounding protocol. The short notation $\mathcal{V}^{NA}$ denotes the exact running of the reader protocol $\mathcal{V}$ without the verification steps and the (generation of the) authentication bit. By $\mathcal{V}^A(\mathsf{st}_{\mathcal{V}})$ we denote the run of the verification steps as in $\mathcal{V}$ for the secret key stored in $\mathsf{st}_{\mathcal{V}}$; we also write $b \leftarrow \mathcal{V}^A(\mathsf{st}_{\mathcal{V}})$ to denote that the verification output is a bit b.

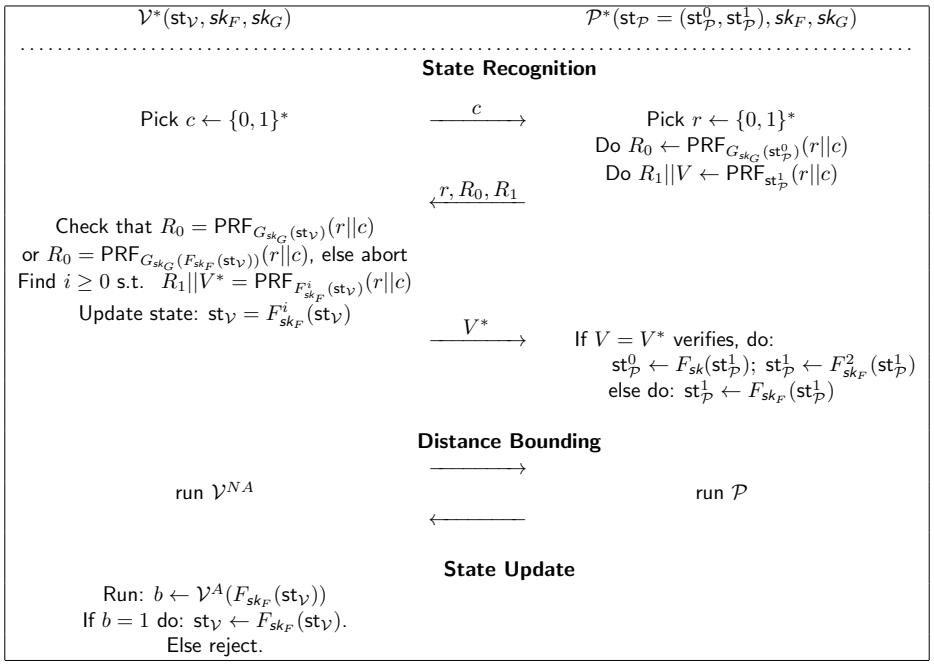

Figure 17: Generic compiler: preservation of impersonation security, mafia, and distance fraud resistance

This compiler is a modification of the narrow-destructive authentication protocol due to Vaudenay [73]. Note that the

distance-bounding protocol is run with $\mathsf{st}_{\mathcal{P}}^1$, which is the current iteration of the secret key. As in [73], the tag updates this state at every distance-bounding attempt; however, in the narrow-destructive protocol in [73], the reader does not update state at all. By contrast, we allow the reader to update state, but only if the distance-bounding protocol succeeds.
We briefly give an intuition why the distance-bounding properties of the underlying protocol are preserved before giving a formal proof of the security statements. We also discuss why terrorist fraud resistance is *not* achieved.

PRF output length. The output length of the PRFs considered in our compiler are different. In particular, the PRF used for the computation of $R_1||V$, namely PRF, should be such that the output values R_1 and V both ensure sufficient security. The output length of F and G must be equal to that of the secret key K.

Insight: the tag state. As aforementioned, the tag keeps a state consisting of: (a) an initial state $\mathsf{st}_{\mathcal{P}}^0$, which is not used during distance bounding; (b) the current state $\mathsf{st}_{\mathcal{P}}^1$, used during distance bounding. Intuitively, the second component of the state ensures that the distance-bounding protocol is run with a new key at every attempt; thus, corrupting the tag and obtaining the state will not enable the adversary to link authentication sessions. By contrast, the first component of the state is only updated if, during the initial state recognition phase, the reader authenticates by means of V. Note that V is computed from $c||r$, rather than $r||c$, thus the adversary cannot just replay the values sent by the tag in the same session.
Thus, the first part of the tag state records the last state shared by an honest reader and the honest tag. However, this value cannot be directly used in computation; else, an adversary is able to link distance-bounding sessions using this key. Thus, the tag first transforms $\mathsf{st}_{\mathcal{P}}^0$ by means of the pseudorandom function G. Since we assume that the long-term secret key sk_G, the implementation of G, and any intermediate computation are not leaked at corruption, this ensures that an adversary corrupting a tag cannot link $\mathsf{st}_{\mathcal{P}}^0$ in a past session. Furthermore, since the adversary does not learn sk_F, it cannot learn the value i such that $\mathsf{st}_{\mathcal{P}}^1 = F_{sk_F}^i(\mathsf{st}_{\mathcal{P}}^0)$.
Finally, we explain the reason why $\mathcal{V}$ checks the value of R_0 for either $G_{sk_G}(\mathsf{st}_{\mathcal{P}}^0)$ or for $G_{sk_G}(F_{sk_F}(\mathsf{st}_{\mathcal{P}}^0))$. Assume that initially the reader and tag share state $\mathsf{st}_{\mathcal{P}}^0$. Now, if the protocol runs honestly, then at the end the reader will be on state $\mathsf{st}_{\mathcal{V}} = F_{sk_F}(\mathsf{st}_{\mathcal{P}}^1)$, while the tag will have state $\mathsf{st}_{\mathcal{P}}^0 = F_{sk_F}(\mathsf{st}_{\mathcal{P}}^1)$ and $\mathsf{st}_{\mathcal{P}}^1 = F_{sk_F}^2(\mathsf{st}_{\mathcal{P}}^1)$, thus once again the reader and tag share the first part of the state. If a MITM adversary drops message V, then with high probability the tag has state $\mathsf{st}_{\mathcal{P}}^0 = \mathsf{st}_{\mathcal{P}}^0$ and $\mathsf{st}_{\mathcal{P}}^1 = F_{sk_F}(\mathsf{st}_{\mathcal{P}}^1)$, while the reader does not update state (unless a successful mafia fraud and impersonation attack is run). In this case, again $\mathsf{st}_{\mathcal{P}}^0$ is shared between the prover and the verifier. Finally, if an adversary drops all the messages after V, then the reader updates state to $\mathsf{st}_{\mathcal{V}} = \mathsf{st}_{\mathcal{P}}^1$, while the tag updates state to $\mathsf{st}_{\mathcal{P}}^0 = F_{sk_F}(\mathsf{st}_{\mathcal{P}}^1)$ and $\mathsf{st}_{\mathcal{P}}^1 = F_{sk_F}^2(\mathsf{st}_{\mathcal{P}}^1)$. In this case, at the following authentication, if the adversary drops R_0, R_1, then neither party updates state, while if the adversary allows the values to go to the verifier, then it updates state again to the new $\mathsf{st}_{\mathcal{P}}^1$, since the verifier also checks the validity of R_0 for $F_{sk_F}(\mathsf{st}_{\mathcal{V}})$.

Pseudorandomness of K. Once we introduce key updates, we have to ensure that the distance-bounding protocol still preserves its properties for the pseudorandom keys generated by iterating the PRF F. Thus, the updated keys must be indistinguishable from each other *and from the original key*. Note that it is possible to have a protocol that is mafia, distance, and impersonation resistant if the key generated by Kg is $K = \mathbf{0}^n$, i.e. the all-zero vector of dimension n, but not for another key. Thus, the initial instance of $(\mathsf{Kg}, \mathcal{V}, \mathcal{P})$ (for state K) is mafia, distance, and impersonation resistant, but no other instances are resistant to these attacks (as the state is updated).

Mafia Fraud Resistance. The protocol (Kg, $\mathcal{V}$, $\mathcal{P}$) is mafia fraud resistant. Since all the states, i.e. keys, are indistinguishable, an adversary against the mafia fraud resistance of the modified scheme $(\mathsf{Kg}^*, \mathcal{V}^*, \mathcal{P}^*)$, the adversary roughly succeeds in impersonating to $\mathcal{V}^*$ for one particular $\mathsf{st}_{\mathcal{P}}^1$ as it does for another. During the reduction, the adversary will simply have to guess which of the (polynomially many) states will be used for authentication. Here the information contained by $\mathsf{st}_{\mathcal{P}}^0$ is irrelevant.

Distance Fraud Resistance. This property is trivially preserved, as distance fraud adversaries know all the correct states, which are indistinguishable from one another (thus one session of distance fraud is as easy to attack as another).

Impersonation security. As for mafia fraud resistance, impersonation security is preserved because an adversary against the stateful protocol is as likely to succeed in impersonating the tag for one state as for another. The level of impersonation security is in fact increased, due to the initial authentication step, i.e. the state recognition.

Why Terrorist Fraud Resistance does not Work. Mafia and impersonation resistance are preserved despite the key update since the adversary has no inside information about the updating process. Thus, the adversary has only an outside view on the keys (which are indistinguishable from one another). In the mafia and impersonation fraud proofs, we argue that, except with negligible probability, an adversary learns as much information for one state as he does for another. For distance fraud, the adversary is the tag itself. As the keys are, except with negligible probability, indistinguishable from each other, the adversary has as much probability to succeed in a single instance of the secret key as it does for multiple keys.

For terrorist fraud resistance, the adversary is at neither of the two extremes, i.e. it *may* learn some insider information about the states and about the updating process, but it does *not* have complete information about it (else, it can then authenticate without the malicious tag). In Chapter 2, we define terrorist fraud resistance in terms of a simulator. Informally, once an adversary having offline contact with a malicious tag succeeds, the simulator also gets as many attempts as the adversary to authenticate to the reader. The simulator, however, only has access to the adversary's transcripts. A protocol is terrorist fraud resistant if for any adversary that succeeds with probability $p_{\mathcal{A}}$, there exists a simulator $\mathcal{S}$ that, once the adversary is successful, runs as many impersonation attempts as $\mathcal{A}$ and wins with probability $p_{\mathcal{S}} \geq p_{\mathcal{A}}$. Intuitively, the information given by the malicious tag to the adversary not only helps it authenticate in a specific impersonation attempt, but the adversary can then also authenticate without the tag's help with at least as much probability (for more details, see Section 2.1).

Our compiler does not preserve terrorist fraud resistance in a provable way, because the malicious tag could reveal some partial information about the secret key *sk* (though not the entire key), thus giving the adversary some insight for a particular state, but not for others; thus the simulator cannot authenticate with equal probability afterwards. It seems therefore hard to find a compiler that preserves terrorist fraud resistance for *all* distance-bounding protocols.

We also note that in general, terrorist fraud resistant constructions must provide a back door for the simulator (since we want *provable* terrorist fraud resistance in a very strong sense, and the simulator must account for all possible malicious-tag-strategies). This back door, however, may be inefficient to achieve in practice. Furthermore, the protocol may lose some of its mafia and distance fraud security (since the back door could be used by the adversary in, say, distance fraud). We leave it an open question to achieve a generic compiler that preserves terrorist fraud resistance for distance-bounding protocols. A related question is whether we can achieve a compiler that preserves terrorist fraud resistance in the independent, game-based notion we present in Chapter 5.

Bypassing impossibility. Paise and Vaudenay proved in [64] that it is impossible to achieve narrow-forward privacy with mutual authentication. However, it seems that our protocol does achieve mutual authentication, thus we cannot achieve narrow-forward privacy. Since Vaudenay [73] shows that narrow-destructive privacy implies narrow-forward privacy, this would imply that our protocol cannot be narrow-destructive private.

However, our protocol does not achieve mutual authentication. In particular, an adversary can always hijack an authentication session after the tag receives V from the reader. Similarly, an adversary can also drop the last authentication message sent by the reader, thus the reader can authenticate the tag, but the tag won't authenticate the reader. Thus, we bypass the impossibility result by not achieving mutual authentication; however, this is not necessary in order to achieve narrow-destructive privacy.

3.2.2 Compiler Properties

Theorem 3.3 *Let* $\mathsf{DB} = (\mathsf{Kg}, \mathcal{V}, \mathcal{P})$ *be a distance-bounding protocol for timing parameters* $(t_{\max}, N_c, E_{\max}, T_{\max})$*, with the restriction that* Kg *outputs only pseudorandom keys. Let* $\mathsf{DB}^* = (\mathsf{Kg}^*, \mathcal{V}^*, \mathcal{P}^*)$ *be the distance-bounding protocol obtained by running the compiler in Figure 17 on* DB*. The following statements hold:*

Mafia Fraud. *For every* $(t^{mafia}, q_{\mathcal{V}}^{mafia}, q_{\mathcal{P}}^{mafia}, q_{\mathrm{OBS}}^{mafia}, \epsilon^{mafia})$*-adversary* $\mathcal{A}^*$ *against the mafia fraud of* DB^* *there is an adversary* $\mathcal{A}$ *against the mafia fraud of* DB*, running in time* $\mathcal{O}\left(t^{mafia}\right)$ *running: no eavesdropping sessions against* DB*, at most* 1 *session with the tag, and at most* $q_{\mathcal{V}}^{mafia}$ *sessions with the reader, and winning with probability of at least (up to negligible terms)* $\frac{1}{q_{\mathcal{P}}^{mafia} \cdot q_{\mathrm{OBS}}^{mafia}} \epsilon^{mafia}$*.*

Distance Fraud. *For every* $(t^{dist}, q_{\mathcal{V}}^{dist}, \epsilon^{dist})$*-adversary* $\mathcal{A}^*$ *against the distance fraud of* DB^* *there is an adversary* $\mathcal{A}$ *against the distance fraud of* DB*, running in time* $\mathcal{O}\left(t^{dist}\right)$ *running at most* $q_{\mathcal{V}}^{dist}$ *sessions with the reader, and winning with probability of at least (up to negligible terms)* ϵ^{dist}*.*

Impersonation Security. *For every* $(t^{imp}, q_{\mathcal{V}}^{imp}, q_{\mathcal{P}}^{imp}, q_{\mathrm{OBS}}^{imp}, \epsilon^{imp})$*-adversary* $\mathcal{A}^*$ *against the impersonation security of* DB^* *there is an adversary* $\mathcal{A}$ *against the impersonation security of* DB*, running in time* $\mathcal{O}\left(t^{imp}\right)$ *running: no eavesdropping*

sessions against DB*, at most* 1 *session with the tag, and at most* $q_{\mathcal{V}}^{imp}$ *sessions with the reader, and winning with probability of at least (up to negligible terms)* $\frac{1}{q_{\mathcal{P}}^{imp} \cdot q_{\text{OBS}}^{imp}} \epsilon^{imp}$.

AVAILABILITY. *For every* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}}, \epsilon)$*-adversary* $\mathcal{A}^*$ *against the availability of* DB* *there exist: adversaries* $\mathcal{A}^{\mathsf{PRF}}, \mathcal{A}'^{\mathsf{PRF}}$ *against the pseudo-randomness of* G*; an adversary* $\mathcal{A}^{imp}$ *against the impersonation fraud of* DB*; and an adversary* $\mathcal{A}^{mafia}$ *against the mafia-fraud resistance of* DB *such that:*

$$\begin{aligned} &\mathbf{Adv}_{\mathsf{PRF}}^{\mathsf{PR}}(\mathcal{A}^{\mathsf{PRF}}) \cdot \mathbf{Adv}_{\mathsf{DB}}^{imp}(\mathcal{A}^{imp}) \cdot \mathbf{Adv}_{\mathsf{DB}}^{mafia}(\mathcal{A}^{mafia}) \\ +q_{\mathcal{P}}(q_{\text{OBS}} + q_{\mathcal{V}}) \cdot 2^{-|V|} &+ \mathbf{Adv}_{\mathsf{PRF}}^{\mathsf{PR}}(\mathcal{A}'^{\mathsf{PRF}}) + q_{\mathcal{V}} \cdot 2^{-|R_1|} + \mathbf{Adv}_{\mathsf{PRF}}^{\mathsf{PR}}(\mathcal{A}^{\mathsf{PRF}}) \quad \geq \quad \epsilon. \end{aligned}$$

Here, $\mathbf{Adv}_{G}^{\mathsf{PR}}(\mathcal{A}^{\mathsf{PRF}})$ *resp.* $\mathbf{Adv}_{G}^{\mathsf{PR}}(\mathcal{A}'^{\mathsf{PRF}})$ *are: the advantage of* $\mathcal{A}^{\mathsf{PRF}}$*, resp.* $\mathcal{A}'^{\mathsf{PRF}}$ *against the pseudorandomness of* G*, while* $\mathbf{Adv}_{\mathsf{DB}}^{imp}(\mathcal{A}^{imp})$*, resp.* $\mathbf{Adv}_{\mathsf{DB}}^{mafia}(\mathcal{A}^{mafia})$ *are the advantages of* $\mathcal{A}^{imp}$*, resp.* $\mathcal{A}^{mafia}$ *to win against the impersonation security, resp. the mafia fraud resistance of* DB*.*

PRIVACY. *Assuming that* DB *is narrow-weak private, the compiled scheme is narrow-destructive private in the sense of Vaudenay.*

Proof. We proceed to prove that the protocol is mafia, distance, and impersonation secure, available, and narrow-destructive private.

Mafia Fraud Resistance. Assume that there exists a $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}})$-mafia fraud adversary $\mathcal{A}^*$ winning against the compiled protocol DB* with probability ϵ. We construct an adversary $\mathcal{A}$ against the mafia fraud resistance of DB that wins with probability at least $\frac{1}{q_{\text{OBS}} q_{\mathcal{P}}} \epsilon + \mathbf{Adv}_{F}^{\mathsf{PR}}(\mathcal{A}^*) + \mathbf{Adv}_{\mathsf{Kg}}^{\mathsf{PR}}(\mathcal{A}^*)$, where $\mathbf{Adv}_{F}^{\mathsf{PR}}(\mathcal{A}^*)$, resp. $\mathbf{Adv}_{\mathsf{Kg}}^{\mathsf{PR}}(\mathcal{A}^*)$ are the distinguishing advantage against the PRF F, resp. the output of Kg. Firstly, we note that the state $\mathsf{st}_{\mathcal{P}}^0$ does not play a role in this proof, since it is not actually used during for distance bounding; the only value that depends on $\mathsf{st}_{\mathcal{P}}^0$ is R_0, which is sent during the initial lazy state-recognition phase. We can thus disregard it. Note that the game $\mathcal{A}$ plays against DB only uses a single state; thus for our reduction, the adversary $\mathcal{A}$ has to guess when adversary $\mathcal{A}^*$ makes its successful impersonation attempt.
In particular, $\mathcal{A}$ must guess which state $\mathsf{st}_{\mathcal{P}}^1$ the tag $\mathcal{P}^*$ and the reader $\mathcal{V}^*$ share when $\mathcal{A}^*$ succeeds in its impersonation attempt. For this state, $\mathcal{A}$ answers all of $\mathcal{A}^*$'s queries by forwarding them to $\mathcal{V}$ and resp. $\mathcal{P}$ (note that the initial state recognition is done in a lazy phase, thus it can be simply forwarded by the adversary $\mathcal{A}^*$). For all other states, $\mathcal{A}$ simulates the reader and tag protocols for a randomly chosen key (generated honestly through Kg). This simulation does not significantly affect $\mathcal{A}^*$'s success probability, due to the pseudorandomness of F, resp. of the keys output by Kg. In fact, $\mathcal{A}^*$'s success probability only decreases by $\mathbf{Adv}_{F}^{\mathsf{PR}}(\mathcal{A}^*) + \mathbf{Adv}_{\mathsf{Kg}}^{\mathsf{PR}}(\mathcal{A}^*)$ accounting for the distinguishing advantage against F and resp. the output of Kg.
In order to guess the shared state for the successful impersonation, $\mathcal{A}$ needs to guess exactly (a) how many reader-tag sessions $\mathcal{A}^*$ runs before the successful authentication (since reader-updates sessions make both $\mathcal{V}^*$ and $\mathcal{P}^*$ update state) and (b) how many successful adversary-tag sessions $\mathcal{A}^*$ runs after the last reader-tag session and before its successful impersonation (since adversary-tag sessions could, depending on the underlying protocol, change the tag's state making the reader catch up to a different state). Now $\mathcal{A}$ guesses both these values with probability $\frac{1}{q_{\text{OBS}}^{\mathsf{mafia}} q_{\mathcal{P}}^{\mathsf{mafia}}}$. If $\mathcal{A}^*$ initiates another reader-tag session or another adversary-tag session before successfully authenticating, $\mathcal{A}$ outputs $\perp$ and halts (it fails). This happens if $\mathcal{A}$ has guessed either q_o or q_t incorrectly. During session q_t (which is an adversary-tag session), $\mathcal{A}$ must simulate the environment for $\mathcal{A}^*$. Upon receiving the challenge c, the adversary forwards values r, R_0, R_1 at random (this affects $\mathcal{A}^*$'s success probability by at most the distinguishing advantage against PRF) and during the distance-bounding phase, $\mathcal{A}$ forwards $\mathcal{A}^*$'s queries to $\mathcal{P}$ and then forwards $\mathcal{P}$'s responses. For every reader-adversary session that $\mathcal{A}^*$ initiates (note that there can be at most $q_{\mathcal{V}}^{\mathsf{mafia}}$ such sessions), $\mathcal{A}$ opens a reader-adversary session and queries $\mathcal{V}$ as $\mathcal{A}^*$ queries $\mathcal{P}^*$. Now $\mathcal{A}^*$ wins with probability negligibly close to $\frac{\epsilon^{\mathsf{mafia}}}{q_{\text{OBS}} q_{\mathcal{P}}}$ in one of the maximum $q_{\mathcal{V}}^{\mathsf{mafia}}$ reader-adversary sessions, and so does $\mathcal{A}$. This yields the bound above for $\mathcal{A}^*$.

Impersonation security The same applies for impersonation security, except that (intuitively) $\mathcal{A}$ also gains some security in future impersonation attempts (for the initial state recognition phase). The reduction works as before, with $\mathcal{A}$ simulating all but one session for $\mathcal{A}^*$.

Distance fraud resistance This statement follows trivially: since distance fraud adversaries are malicious tags, they know the secret keys resp. states. Since the outputs of Kg are pseudorandom, the distance fraud level just transfers trivially. Note, however, that in view of our results in Chapter 2, few protocols are, actually, distance-fraud resistant. Our compiler does not create distance-fraud resistance, but rather preserves it if the protocol is already distance-fraud resistant.

Privacy This proof follows the lines of the proof of theorem 15 in [73], with the following changes: (1) up to a negligible difference ($\mathbf{Adv}^{\mathsf{PR}}_{F}(\mathcal{A}^*) + \mathbf{Adv}^{\mathsf{PR}}_{\mathsf{Kg}}(\mathcal{A}^*)$) we disregard the key update step and assume that the same key is being used (this is possible since corrupted tags are destroyed, and since states are pseudorandom); thus we reduce destructive privacy to the narrow-weak privacy of the underlying protocol, thus (2) replacing the random c output by the blinder in simulated SendTag queries against DB* by the responses given by the narrow-weak private blinder against DB. Under the assumption of (1) this is a perfect simulation of SendTag queries. The rest of the proof is the same as in [73]. Note that this proof is only possible since we assumed that the adversary does not learn the implementations of F and G, and since the keys sk_F and sk_G are not leaked to the adversary.

Availability The scheme DB* is available under the assumption of completeness for DB. In this setting, adversaries can run reader-tag sessions where they observe communication, can interact either with the reader or with the tag separately, or they can run MITM attacks (even using pure relay). Note that an adversary can only break the availability of the scheme in three ways: (1) if the reader updates state but the tag does not; (2) the tag updates $\mathsf{st}^0_{\mathcal{P}}$, but the reader does not update $\mathsf{st}_{\mathcal{V}}$; (3) the reader identifies the "wrong" state from R_1. The latter only happens if the adversary comes up with a valid R_1 value, which it does only with probability $q_{\mathcal{V}} \cdot 2^{-|R_1|} + \mathbf{Adv}^{\mathsf{PR}}_{\mathsf{PRF}}(\mathcal{A}^{\mathsf{PRF}})$. Case (2) only happens if the adversary can send a correct authentication string V. We show that this happens only with probability $q_{\mathcal{P}}(q_{\text{OBS}} + q_{\mathcal{V}}) \cdot 2^{-|V|} + \mathbf{Adv}^{\mathsf{PR}}_{\mathsf{PRF}}(\mathcal{A}'^{\mathsf{PRF}})$. Indeed, an adversary here can either (a) hope that the values of c and r are a replay for the same key $\mathsf{st}^0_{\mathcal{P}}$ in some other session where either both c and r, or just r, were generated honestly; (b) in case the adversary can simply guess V with non-negligible probability, then this adversary can be used to distinguish the output of PRF from V. This leads to the above probability.
We now examine case (1). A reader-tag session will make both reader and tag update state; thus this will not help the adversary. If the adversary runs an adversary-tag session, the tag may update state, but the reader does not. If the adversary runs a reader-adversary session (without running a parallel adversary-tag session), if the adversary fails to authenticate, then the reader does not update state.
The adversary $\mathcal{A}$ can only win in a reader-adversary session, which can be run either in a MITM attack or in a separate reader-adversary interaction. If the adversary runs a MITM attack and forwards the tag a challenge (be it the reader's challenge or another challenge), the tag updates state, and thus the adversary fails. Therefore, the adversary only wins if it wins in a reader-adversary interaction without querying the tag. Thus, the adversary must (i) pass the initial state recognition phase; (ii) authenticate during distance bounding. In order to achieve step (ii) the adversary must break both the impersonation security and the mafia-fraud resistance of the underlying distance-bounding protocol. The adversary passes the initial state recognition only if it can break the pseudorandomness of G. Thus we have the indicated bound. □

3.2.3 Optimisations

No Mutual Authentication. The initial state update computation in the compiler (which is quite computationally expensive) is required by protocols with no partial reader-authentication, as e.g. the protocol in [49]. However, many distance-bounding protocols, e.g. [6, 42] do not feature reader authentication. In this case, the compiler can be simplified as in Figure 18. This compiler can be used if the underlying bounding protocol has properties (1-4) as in Theorem 3.3, and (5) the partial reader protocol $\mathcal{V}^{NA}$ is independent of the state $\mathsf{st}_{\mathcal{V}}$ of the reader. Protocols fulfilling this conditions are, e.g. [6, 13, 42].

Merging Authentication Steps. Even if some reader authentication is used, we can gain efficiency by merging the state recognition step with impersonation security. Some distance-bounding schemes, e.g. [6], have some PRF-based lazy-phase authentication preceding the time-critical rounds. In particular, we require that the underlying scheme DB has phases $1, \ldots, n$ such that there is a (lazy) phase l where $\mathcal{P}$ sends a (part of the) output of a PRF function G^* computed on state information $\mathsf{st}^1_{\mathcal{P}}$ (for static protocols, this can be the key generated by Kg) to the reader *and* there is no phase $1, \ldots, l-1$ in $\mathcal{V}^{NA}$ that depends on $\mathsf{st}_{\mathcal{V}}$. Then, we tweak the compiler using the PRF output R_1 for tag authentication.

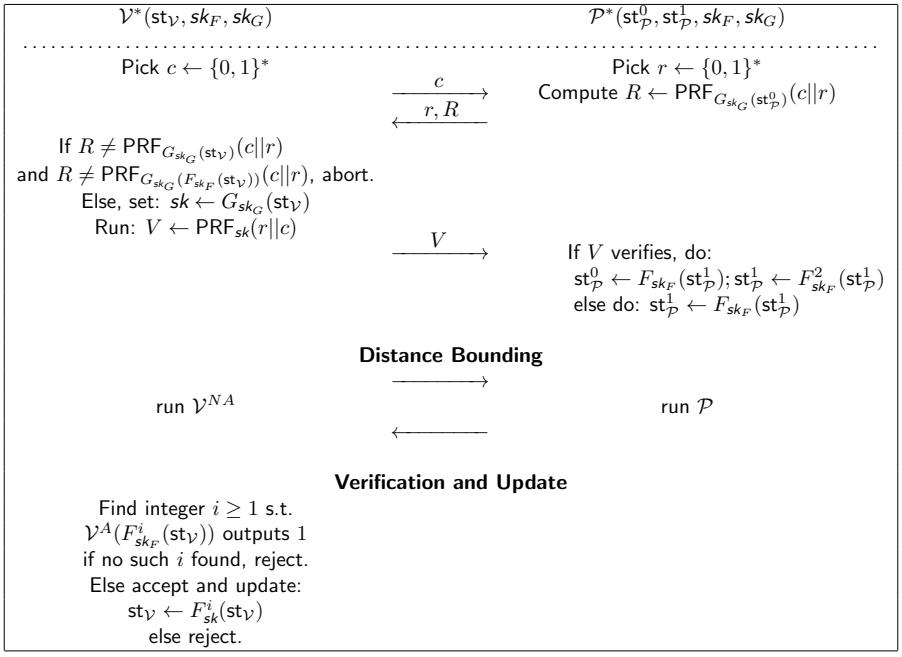

Figure 18: Compiler for protocols with no mutual authentication

4 In-Depth: Mafia Fraud Insights

The framework we define in Chapter 2 introduces the following attacks against distance-bounding protocols: (1) mafia fraud, where the attacker runs a MITM attack in the presence of an honest prover and an honest verifier (the verifier's clock will detect *pure relaying* though); (2) terrorist fraud, where the adversary receives limited offline support from the dishonest prover such that the adversary authenticates (however, the prover's aid should not allow the adversary to authenticate by itself) ; (3) distance fraud, where the adversary is the dishonest tag itself and wants to authenticate from outside the verifier's proximity; (4) classical lazy-phase impersonation, where the adversary tries to impersonate a legitimate prover during the lazy phases of the protocol.

One particularly-subtle form of mafia fraud is what we call a key-learning attack (denoted KLMF), where an adversary uses an interdependency between the time-critical responses computed by both prover and verifier in order to learn a long-term secret key. By subsequently using this key, the adversary is able to authenticate. We show an example of such an attack in our analysis of the protocol due to Reid et al. in section 2.3.5. Key-learning attacks are especially effective against protocols aiming to achieve terrorist fraud resistance. This is because most terrorist fraud resistant schemes need to relate the two time-critical responses such that, if both responses are forwarded from the dishonest prover to the terrorist adversary, the adversary will learn the long-term secret.

In particular, in many distance-bounding schemes that strive towards lightweight computation, such as the schemes due respectively to Bussard and Bagga [15], Reid et al. [70], and Kim et al. [50], the prover and verifier compute two response strings, such that the bitwise XOR sum of the two values yields a long-term secret. In the time-critical phases, the verifier forwards a challenge consisting of a single bit, and receives a response from one of the two response strings. Against such schemes, a key-learning attack that succeeds with probability 1 is structured as follows: the adversary first forwards a number of time-critical response between the prover and verifier (essentially relaying the information), then, in a single round, it forwards the response bit from the wrong response string. If the verifier then accepts, the adversary knows the bit he forwarded was equal to the bit that the verifier expected (i.e. the bit from the other string). Since the XOR of the two bits is a bit of a long-term secret, the adversary then knows that this bit is 0. Else, if the verifier rejects, the adversary knows that the bit is 1. Thus, by learning the key bit-by-bit, the adversary is then able to impersonate an out-of-range prover, without the prover's knowledge or consent.

In order to visualize this attack in practice, recall the example of Alice and Bob from Chapter 2. There, Alice owns the unique pass key to a locker, and is away one evening. In this setting Bob tries to authenticate at the gym and get access

to the locker. For key-learning (KLMF) attacks, Bob eavesdrops and actively interferes with several runs between Alice and the locker, effectively flipping a few bits and learning the bits of the key. Then Bob waits for an evening when Alice is *away* to use this knowledge. This attack is implementable in practice and is captured by our model in Section 2.
In KLMF attacks, the adversary does *not* authenticate in a session where the prover and verifier have had any honest exchanges (though the adversary runs a MITM strategy with the distant prover during authentication). The effectiveness of this attack depends on the adversary's ability to extract information about the key by interfering in honest reader-tag sessions *and* learning whether the authentication has succeeded or not. In particular, key learning does not work against most distance-bounding protocols (which do not aim to counter terrorist fraud attacks); in such protocols, there is no correlation between the time-critical responses. Furthermore, such attacks do not work when the authentication result does not depend solely on the flipped response-bits. In particular, we are able to prove that the scheme due to Kim et al. is in fact mafia fraud resistant, due to a second authentication phase. Essentially in this protocol, the adversary can still flip bits in time-critical phases; however, doing so will with great probability make the verifier reject the prover regardless of the correlation between the time-critical response bits, since the adversary is unable to come up with an authentication string that verifies at the end of the execution.
In key-learning attacks, the adversary first learns the key and then uses this knowledge in a separate session, while the prover is away. As an extension of this attack and of mafia fraud in general, we can also consider *strong mafia fraud* (strMF), where the adversary can *continue* an aborted prover-verifier session and thus authenticate. In the example of Alice and the pass key, Bob may now also try to access the locker while Alice *is* at the gym, e.g. if she gets called away after having started her authentication.
This latter attack falls outside the model we showed in Chapter 2, since it assumes that the adversary may taint more than the allowed number of rounds during the final authentication phase. In fact, this attack is not a usual mafia fraud attack in the sense that, during classical mafia fraud, the prover needs to be away from the verifier for the entire authentication attempt. Here, we assume that the prover only needs to be away from the verifier for a part of the authentication session. In fact, most protocols are vulnerable to strong mafia attacks if the prover aborts after a large number of rounds. We point out that by, e.g., side-channel attacks or by interfering with wireless transmissions during rapid bit exchange, the adversary could, in practice, mount key-learning attacks and strong mafia attacks. It thus makes sense to both extend the mafia fraud model such that strong mafia fraud is formally captured, and to design protocols that are secure in the presence of aborts. The work we present in this chapter is outlined in [61].

Contributions. We first extend our definition of mafia fraud to capture strong mafia attacks (resp. strMF security), where the adversary can simply take over from an honest prover in its authentication attempt. In particular, we now allow the adversary to win in a reader-adversary session sid also if it taints more than $T_{\max}$ rounds, with the stipulation that there exists a number $i \in \{1, \ldots, N_c\}$ such that the adversary communicates with no other adversary-tag session from round i until round N_c.
We also show that this model is strictly stronger than mafia fraud resistance. Finally, we show a compiler to obtain strMF resistance from mafia fraud resistance. There are two tricks we use for our compiler: (1) ensure that the adversary cannot simply take over from the aborting prover by ensuring that the last response sent by the prover is (part of) the output of a pseudo-random function run on two newly-generated nonces and the challenges and responses exchanged during the time-critical phases; (2) this second PRF uses a different key, independent of the key used for the underlying distance-bounding protocol. Since the adversary has only negligible probability of guessing the authentication value, it also has a negligible probability of winning in a strMF attack against the compiled protocol. We illustrate the use of our compiler on the Hancke and Kuhn protocol [42], which we turn into a strMF-resistant construction.
Our strategy is thus as follows:

- we extend the definition of mafia fraud to include so-called strong mafia fraud attacks (strMF), and show that strMF security is a strictly stronger property than mafia fraud resistance.
- we show a compiler that takes as input a mafia fraud resistant distance-bounding protocol; the compiler then yields a protocol that is strMF secure and which also preserves the remaining properties of the original protocol, apart from terrorist fraud resistance. Essentially, we ensure that the adversary cannot complete a hijacked reader-tag session, since an added final round of lazy authentication is nearly impossible to pass without help from the honest prover.
- we run the strMF compiler on the Hancke and Kuhn protocol, thus obtaining a strMF secure construction.
- we show that our compiler still works for protocols that are not entirely mafia fraud resistant, but are vulnerable to a single type of mafia fraud attacks, namely key-learning mafia fraud (KLMF). This is ensured since the last authentication phase also protects the protocol from attacks where the adversary tries to flip some bits in the communication in order to learn a long-term secret key.

4.1 Strong Mafia Fraud Resistance

We begin by recalling that in the basic communication model of the framework in Chapter 2 the adversary may have three types of interactions with the prover (tag) or the verifier (reader): eavesdropping an entire session (in reader-tag sessions), impersonating the reader to the tag (in adversary-tag sessions), or impersonating the tag to the reader (in reader-adversary sessions). In the context of mafia fraud attacks, the adversary can use these interactions arbitrarily; the aim of the attack is for the adversary to authenticate to the verifier without purely relaying too many time-critical phases (during that particular authentication attempt). In particular, pure relay (forwarding exact messages between the reader and the tag, and using relaying scheduling) taints time-critical phases; depending on the protocol, a small number $T_{\max}$ of time-critical phases may also be purely relayed between the verifier and a distant prover during a communication attempt (this accounts for possible delays in transmission, across an unreliable channel).
In the definition of mafia fraud resistance, it is specified that an adversary may not taint more than $T_{\max}$ phases of the reader-adversary session where it authenticates to the verifier. Thus, this model does not allow for sessions where the adversary authenticates in a session where it *mostly relays, and at the end hijacks* communication. Such attacks are effective against most distance-bounding protocols in the literature, and we call them strong mafia fraud. Thus, in the case of strong mafia fraud, an adversary running a reader-adversary and an adversary-tag session in parallel could purely relay a number of rounds (this would result in an honest reader-tag session) and then simply *take over* from the prover in an aborted authentication session. Strong mafia fraud is therefore an extension of mafia fraud, which includes attacks where the adversary can win by authenticating in a reader-adversary session in which pure relay is used for a number of time-critical rounds, after which the adversary may no longer query the prover.

A strMF **attack.** Consider the mafia fraud resistant scheme of Hancke and Kuhn, as presented in Chapter 2. This protocol ends in N_c time-critical phases, where, for each phase the verifier sends a challenge bit and expects a response bit coming from one of two strings, which are output by a pseudorandom function. In the mafia fraud scenario, the security of the protocol relies on the fact that the two response strings are pseudorandom and unrelated; however, if an honest reader-tag session is aborted in one of the last rounds, the adversary can just take over the authentication and has a large probability of guessing the remaining responses. Thus, the same protocol is not strong mafia fraud resistant.

The strMF **model.** The key point to describing the strong mafia attack is to allow the adversary to authenticate to the verifier even during an authentication session started by the honest prover with the honest verifier. In our model, we can capture the adversary's hijacking of some prover-verifier session in terms of parallel sessions, say sid and sid*, such that sid is a reader-adversary session and sid* is an adversary-tag session. We describe the honest interaction between the prover and the verifier as an adversary purely relaying a number of phases between the two sessions (corresponding to the phases run between the prover and verifier); subsequently, the prover ceases the communication, i.e. session sid* is no longer useful, and the adversary must authenticate to the verifier in session sid.
We consider protocols between a prover and verifier consisting of some lazy phases and a number N_c of time-critical phases, in total N of messages *sent by the prover*. We denote each time-critical phase $i \in \{1, \ldots, N_c\}$ as $\Pi^i_{\mathsf{sid}}[k_i \ldots k_i + 2\ell - 1] = (m_{k_i}, \ldots, m_{k_i+2\ell-1})$ for $k_i, \ell \geq 1$, where each time-critical phase consists of 2ℓ messages. We make no assumption regarding the order in which the phases are scheduled in the protocol execution, however.

Definition 4.1 (Strong Mafia Fraud Resistance) *Consider a distance-bounding authentication scheme* DB *with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$*, consisting of* $2N$ *messages in total,* N *sent by the prover and* N *sent by the verifier. A* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}}, q_{\text{EXT}})$-strMF *adversary* $\mathcal{A}$ wins *against* DB *if the verifier accepts in a verifier-adversary session* sid *such that, if there exists an adversary-prover session* sid* *that taints more than* $T_{\max}$ *time-critical phases of* sid*, then it holds that there exists* $i \in \{1, \ldots, N-1\}$ *such that, for all adversary-prover sessions* sid$'$ *tainting more than* $T_{\max}$ *sessions of* sid *it holds that:*

$$\mathit{clock}(\mathsf{sid}, 2i+1) > \mathit{clock}(\mathsf{sid}', 2i)$$

and for all $N \geq j > i$*, it holds that:*

$$\mathit{clock}(\mathsf{sid}, j) < \mathit{clock}(\mathsf{sid}', i+1).$$

Let $\mathbf{Adv}^{\mathsf{strMF}}_{\mathsf{DB}}(\mathcal{A})$ *denote the probability that* $\mathcal{A}$ *wins. We say that* DB *is Strong Mafia Fraud resistant (i.e.* strMF *resistant) if for any adversary* $\mathcal{A}$ *its advantage* $\mathbf{Adv}^{\mathsf{strMF}}_{\mathsf{DB}}(\mathcal{A})$ *is at most negligible in the number* N_c *of time-critical rounds.*

In this definition, the index i denotes the last round that the prover is online; after sending the message indexed i (note that i is strictly smaller than N, thus we require that the prover does *not* send all the messages it must send during the protocol). An adversary wins, thus, if it authenticates to the verifier either in a verifier-adversary session where less than

$T_{\max}$ phases have been tainted, or in a session where more than $T_{\max}$ phases have been tainted such that the prover stops being available after round i of messages. This round could be part of a time-critical phase or it could be a lazy-phase message.

The strMF **Compiler.** We present a general compiler to turn a mafia fraud resistant protocol into a strMF resistant protocol. Our strategy is two-fold: (i) we add a lazy phase after the distance-bounding protocol has been run, in which the prover sends the PRF output of two newly-generated nonces, as well as the messages exchanged during the underlying protocol; and (ii) in the computation of the pseudorandom function, we use a different key from the one used during the underlying protocol, as well as a different pseudorandom function from the ones used by the protocol. In this way, we ensure that the final lazy phase computation does not leak any further information about the long-term secret used during the distance-bounding protocol. Note that the PRF is computed over the transcript of the entire distance-bounding protocol, including the time-critical phases. In particular, we also aim at thwarting key-learning attacks in protocols where the time-critical responses are related.
We assume that the underlying protocol is mafia-fraud resistant. Before the underlying protocol is run, two nonces are exchanged between the prover and the verifier; these nonces are used as part of the input in a final authentication message from the prover to the verifier. Since the nonces are freshly chosen at each session, the final authentication message is also fresh for each session and with great probability the adversary is unable to generate it without the aid of the prover. In particular, in time-critical phases where more than $T_{\max}$ phases have been tainted the prover has aborted, thus the adversary cannot query it in order to learn the final prover response from an adversary-prover session. This is depicted in Figure 19. Note that, although we only assume the underlying protocol to be mafia fraud resistant, the transformation also preserves distance fraud resistance and impersonation security.
We denote by $M_{\mathcal{V}}$ and resp. $M_{\mathcal{P}}$ the nonces generated by the verifier, resp. the prover before the distance-bounding protocol is run. We use a pseudorandom function F, independent of all the pseudorandom functions used by the underlying distance-bounding authentication protocol DB. We use the notation of Chapter 3 and denote by $\mathcal{V}^{NA}$ the underlying distance-bounding protocol run by the verifier, with the exception of the final authentication step. By $b \leftarrow \mathcal{V}^{A}(sk)$ we denote the running of the authentication algorithm of the verifier $\mathcal{V}$ by using the secret key sk, resulting in bit b. Finally, we write τ_{DB} to indicate the transcript of the messages exchanged during the distance-bounding protocol DB in the order they are exchanged.

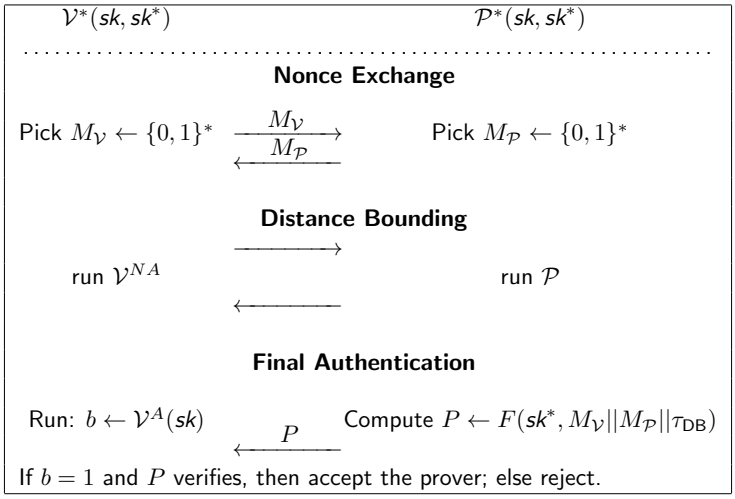

Figure 19: Compiler: achieving strMF security from mafia fraud resistance

The following statement holds:

Lemma 4.2 *Let* DB $= (\mathsf{Kg}, \mathcal{V}, \mathcal{P})$ *be a distance-bounding protocol for timing parameters* $(t_{\max}, N_c, E_{\max}, T_{\max})$. *Let* DB *be* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\mathrm{OBS}}, \epsilon)$ *mafia-fraud-resistant. Let* $\mathsf{DB}^* = (\mathsf{Kg}^*, \mathcal{V}^*, \mathcal{P}^*)$ *be the distance-bounding protocol obtained by*

running the compiler in Figure 19 on DB. *The new scheme* DB* *has (up to negligible terms) at most*

$$q_{\mathcal{V}} \cdot (\epsilon + 2^{-|P|} + \mathbf{Adv}_F^{dist}(\mathcal{A}')) + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|M_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|M_{\mathcal{P}}|}$$

advantage against strong mafia fraud attacks, where $\mathcal{A}'$ is a distinguisher against the pseudorandomness of function F. In this strong mafia fraud attack, the adversary runs $q_{\text{OBS}}, q_{\mathcal{V}}$ resp. $q_{\mathcal{P}}$ sessions. In particular, if DB *is mafia fraud resistant, then* DB* *is* strMF*-secure. We also note, though this is not our focus, that the compiler also preserves distance fraud resistance and impersonation security.*

Proof. For the second part of the theorem, note that the compiler makes no change or addition to the time-critical phases of the protocol; furthermore, the nonces exchanged and the computation of P, using a new PRF and a new key, leak no information about the secrets used during the protocol. Thus the distance-fraud resistance levels are exactly preserved. As far as impersonation security is concerned, we note that the additional lazy impersonation step in fact increases impersonation security (loosely) by a term $q_{\mathcal{V}} 2^{-|P|}$.
We now focus on mafia fraud resistance and strong mafia fraud resistance. First note that the latter notion implies the former notion (this is trivial, following the definition). The proof follows the following main steps: (1) we prove that the nonce pair $(M_{\mathcal{V}}, M_{\mathcal{P}})$ exchanged at the beginning of the compiled protocol is quasi-unique; (2) we reason that, if a successful strong mafia fraud is committed, then either (a) the adversary succeeds in committing mafia fraud against the underlying protocol (and may then forward the value P), or (b) the adversary must guess or forge the value of P in a strong mafia fraud attack.
The first point follows as in the mafia proofs we presented in Section 2.3. Assume that the adversary wins in some verifier-adversary session sid. In particular, the term: $\binom{q_{\mathcal{V}}+q_{\text{OBS}}}{2} \cdot 2^{-|M_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|M_{\mathcal{P}}|}$ accounts for the probability that the nonce pair $(M_{\mathcal{V}}, M_{\mathcal{P}})$ may appear in a different session than sid.
Now we assume that the nonce pair exchanged in session sid is unique. By the definition of strong mafia fraud resistance, there are two ways to succeed in a mafia fraud attack: (a) succeeding to authenticate in session sid having tainted at most $T_{\max}$ time-critical phases; and (b) authenticating in session sid, having tainted more than $T_{\max}$ time-critical phases and succeeding in sending at least the final authentication message on behalf of the prover, without the prover's aid. In the former case, we can construct an adversary against the mafia fraud resistance of the underlying protocol. This accounts for a term ϵ in the success probability of the adversary. Now the adversary has at most probability $2^{-|P|} + \mathbf{Adv}_F^{\text{dist}}(\mathcal{A}')$ to succeed in such an attack, the former term accounting for the adversary's probability to guess P and the latter term accounting for the distinguishing advantage of the adversary $\mathcal{A}'$ against the pseudorandom function F.
Finally, accounting for the fact that there are $q_{\mathcal{V}}$ verifier-adversary sessions, we obtain the stated bound. □

Compiler Optimisations. Note that the compiler first requires both parties to generate fresh randomness; this prevents replay attacks. However, for protocols where both parties already generate fresh randomness (of appropriate length), this step can be skipped. Indeed, this is the approach we take in enhancing the well-known construction of Hancke and Kuhn to withstand strong mafia fraud attacks.

4.2 Application: a strMF-Secure Hancke-Kuhn Protocol

In this section, we show how to run our compiler on the Hancke and Kuhn protocol, described and analysed in Section 2.3.2. The resulting protocol (using the optimisation we suggest above) is shown in Figure 20. In particular, the Hancke-Kuhn protocol is secure against mafia fraud attacks, and also requires both the prover and the verifier to generate fresh randomness at the beginning of the protocol run. In the compiled protocol, the reader and tag share an additional secret key *sk′*, used for the final authentication step. Furthermore, we denote by F the (fresh) pseudorandom function used for the final authentication, and let τ_{DB} denote the concatenation of the values exchanged during the protocol, i.e. $\tau_{\text{DB}} = N_{\mathcal{V}}||N_{\mathcal{P}}||R_0||T_0^{R_0}||\ldots||R_{N_c}||T_{N_c}^{R_{N_c}}$.
We also note that the compiled protocol also gains impersonation security from the state recognition phase: in particular, the original Hancke-Kuhn protocol is not impersonation fraud resistant as it provides no authentication during lazy phases. Furthermore, we note that our compiled protocol assumes that $T_{\max} = 0$, as indeed is the case for the original Hancke-Kuhn protocol; however, it is possible to introduce fault tolerance into the protocol as it was done for the Kim and Avoine construction, outlined in Section 2.3.4.

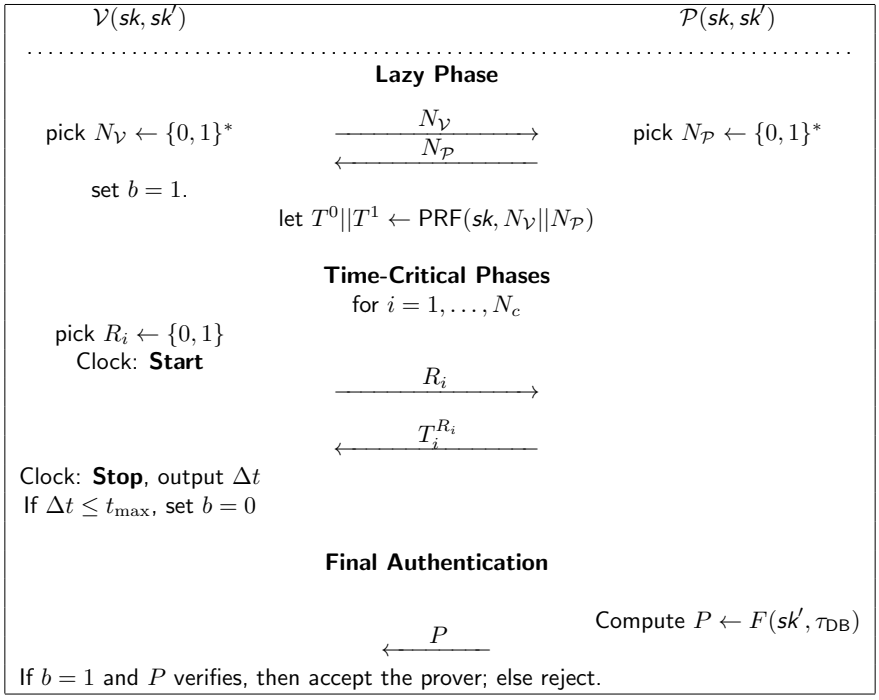

Figure 20: The compiled Hancke and Kuhn protocol

4.3 The Case of Key-Learning Attacks

We note that our compiler requires the underlying distance-bounding protocol to be mafia fraud resistant. However, as we showed in Section 2.3, especially protocols aiming to withstand terrorist fraud attacks are vulnerable to what we call key-learning attacks (KLMF). Note that, since such attacks are a particular type of mafia fraud attacks, any strong mafia fraud resistant protocol is also secure against key learning attacks.

However, there are two issues that we address in this section. These are: (1) can our compiler also turn a protocol that is vulnerable to key learning attacks into a strong mafia fraud secure protocol?; and (2) how can we achieve a mafia fraud resistant (but not necessarily strongly-mafia fraud resistant) protocol out of a construction that is *not* resistance to KLMF attacks? In particular, we use the example of the scheme due to Bussard and Bagga [15] in the remainder of this section.

In what follows we refer to key-learning attacks as specific types of mafia fraud, committed against protocols where the responses that the prover computes, here denoted T^0, T^1, are related such that there exists a function $f_{sk'} : \{0,1\}^{N_c} \longrightarrow \{0,1\}^{N_c}$, depending on a (static) secret key sk' (which we assumed is chosen honestly at random from a distribution $\mathcal{D}$ that is indistinguishable from the uniform random distribution of length depending on a security parameter) such that, for every execution it holds that $T^1 = f_{sk'}(T^0)$. A key learning attack follows a mafia fraud interaction, and it aims to enable the adversary to distinguish the key sk' from random. In practice, the output of the adversary is in fact the entire secret sk' (though we don't require this in our assumptions). More concretely, a key-learning attack requires that the adversary opens in parallel verifier-adversary sessions sid_i and adversary-prover sessions sid'_i, such that sid'_i taints more than $T_{\max}$ time-critical phases of sid_i. Here i denotes an index, which can take arbitrary integer values $i \geq 1$.

We will also refer in the remainder of this section to protocols that are *not* resistant to key-learning attacks, but are otherwise mafia fraud resistant. An example of such a protocol could be the protocol due to Bussard and Bagga depicted below [15].

We note that this protocol has exactly the structure we described above. In particular, the function f depends here on the long-term secret key sk and $T^1 = f_{sk}(T^0) := T^0 \oplus sk$ for any execution of the protocol. The dependency, allowing an adversary to recover the secret sk in case of a terrorist fraud attack, also enables an adversary to learn the secret key bit by bit.

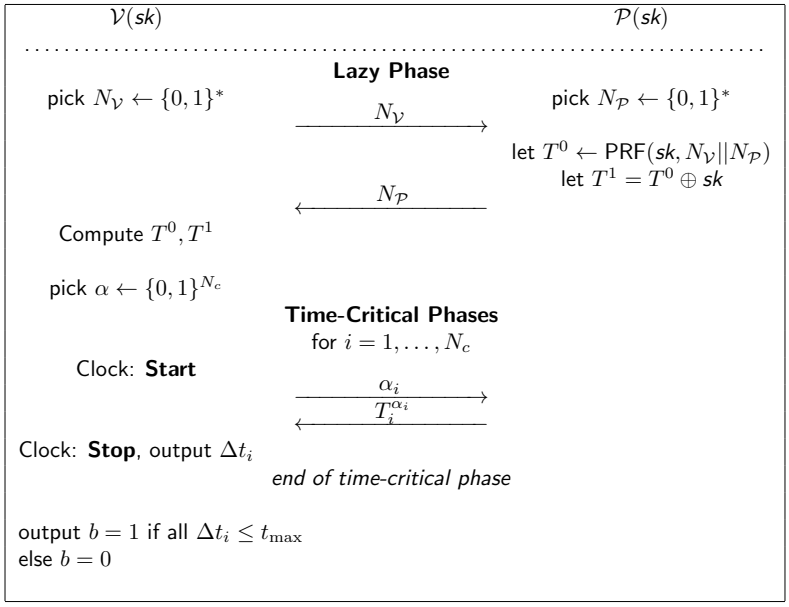

Figure 21: The Bussard-Bagga protocol

A key-learning attack. The key-learning attack we describe in this section follows exactly the attack showed in Section 2.3.5. We recall it briefly in the following. The adversary now aims to compute a guess sk' of the secret key sk; it begins by initialising verifier-adversary sessions sid_i in parallel with adversary-prover sessions sid'_i, where $i = 1, \ldots, N_c$. In each pair of sessions, the adversary first forwards the lazy phase nonces; then, for each index i, the adversary forwards the challenges and responses in all the time-critical phases except phase $N_c - i + 1$. For this phase, the adversary receives some challenge α in session sid_i and forwards its conjugate, i.e. $\bar{\alpha} = \alpha \oplus 1$ in session sid'_i, receiving from the prover some response T, i.e. the corresponding bit from the string $T^{\bar{\alpha}}$. This is the response that the adversary also forwards in session sid_i, and waits to receive the authentication bit from the verifier; if the verifier accepts, than the responses $T^{\bar{\alpha}}$ and T^{α} must be equal, hence the adversary sets the $N_c - i + 1$-th bit of the guess sk' to be equal to 0; else, if the verifier rejects, then the adversary sets the corresponding bit to 1. Finally, at the end of the attack the adversary is able to learn the secret key sk and can impersonate the tag at will.

Strong mafia fraud and key learning. In the previous section we considered a compiler which, on input a mafia fraud resistant distance-bounding protocol, outputs a strong mafia fraud resistant construction. In this section, we show that the protocol need not necessarily be mafia fraud resistant to begin with. In fact, if the protocol is *not* resistant against key learning attacks, but is otherwise mafia fraud resistant, the compiler still grants strong mafia fraud resistance. In particular, thus, the compiled protocol will also be mafia fraud resistant.
Indeed, consider a protocol as described above, which is not resistant to KLMF attacks, but is otherwise mafia fraud resistant. The addition of the final authentication message now thwarts key-learning attacks as we formally proved to be the case for the Swiss Knife protocol (see Section 2.3.6). Essentially, the adversary learns nothing about the key by merely observing protocol runs (since the protocol is secure against all other types of mafia fraud); furthermore, mafia fraud attacks require the effective changing of the messages exchanged in at least one pair of verifier-adversary and adversary-prover session; however, the adversary learns nothing by changing the values, since now with great probability the protocol run with the verifier will fail anyway (the adversary fails to provide legitimate authentication).
We now apply the optimised compiler in the previous section to the Bussard and Bagga protocol, which is vulnerable to key-learning attacks. The compiled protocol is provably strong mafia fraud resistant. We denote by τ_{DB} again the concatenation of the values exchanged during the protocol.

Theorem 4.3 (Bussard-Bagga Properties) *Let* DB *be the distance-bounding authentication scheme in Figure 22 with parameters* $(t_{\max}, N_c)$*, and assume the key sk is pseudorandom and chosen honestly, at random, before tag initialisation*

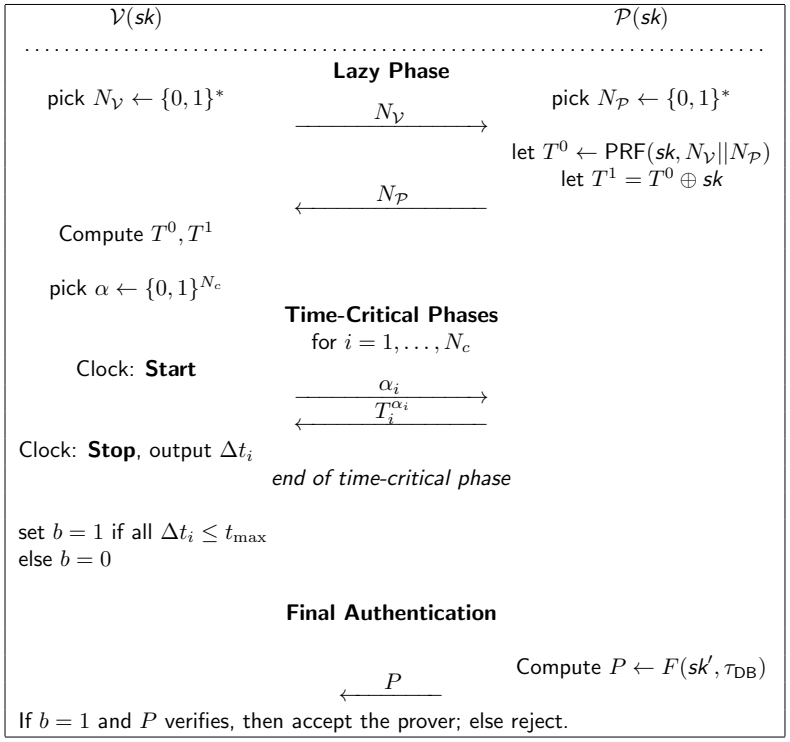

Figure 22: The Compiled Bussard-Bagga protocol

(in particular, the tag does not choose its own key), from a distribution $\mathcal{D}$ which is computationally indistinguishable from the uniform random distribution. This scheme has the following properties:

- *For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-mafia-fraud adversary $\mathcal{A}$ against the scheme there exists a (t', q')-distinguisher $\mathcal{A}'$ against F (where $t' = t + O(n)$ and $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$) such that*

$$\begin{aligned}\mathbf{Adv}^{\textit{mafia}}_{\mathsf{DB}}(\mathcal{A}) \quad \leq \quad & \left(\tfrac{1}{2}\right)^{N_c} + 2\binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-(|N_{\mathcal{V}}| + \lceil \frac{N_c}{2} \rceil)} + 2\binom{q_{\mathcal{P}}}{2} \cdot 2^{-(|N_{\mathcal{P}}| + \lceil \frac{N_c}{2} \rceil)} \\ & + q_{\mathcal{V}} \cdot \mathbf{Adv}^{d}_{F}(\mathcal{A}') + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-(|N_{\mathcal{V}}| + N_c - 1)} + 2\binom{q_{\mathcal{P}}}{2} \cdot 2^{-(|N_{\mathcal{P}}| + N_c - 1)}.\end{aligned}$$

- *For any $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$-strong mafia adversary $\mathcal{A}$ against the scheme, there exists either $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$ mafia fraud adversary $\mathcal{A}'$ against the scheme or there exists an adversary $\mathcal{A}''$ against F (where $t' = t + O(n)$ and $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$) such that*

$$\mathbf{Adv}^{\textit{str.maf.}}_{\mathsf{DB}}(\mathcal{A}) \quad \leq \quad q_{\mathcal{V}} \cdot (\mathbf{Adv}^{\textit{mafia}}_{\mathsf{DB}}(\mathcal{A}') + 2^{-|P|} + \mathbf{Adv}^{\textit{dist}}_{F}(\mathcal{A}'')) + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|M_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|M_{\mathcal{P}}|}.$$

Proof. The proofs for these two statements follow similarly to the proof of mafia fraud resistance for the Swiss Knife protocol (see Section 2.3.6) and resp. the proof for our compiler above. □

Achieving key-learning security. By using the compiler, a protocol that is susceptible to key-learning attacks becomes strong mafia fraud resistant. However, in some scenarios where aborts are not relevant, it may be useful to obtain simply a mafia fraud resistant protocol. There are essentially three ways this can be achieved: (1) remove the dependency between the prover's responses and a long-term secret (this, however, also removes terrorist fraud resistance); (2) update the long term secret *sk* such that $T^1 = f_{sk}(T^0)$ (in this case, though the adversary may learn *sk*, it will not be able to use it for a

subsequent authentication attempt); and (3) ensure, as is the case of the compiler, that the adversary has no direct way of drawing information from performing a key-learning attack.
For the second approach we could for example use the compiler in Chapter 3. Though this approach may also enhance the privacy level of the underlying construction, note that key updates will require greater computational and storage complexity. The most promising approach seems number (3); however, we note that in fact the compiler we present in Figure 19 seems nearly optimal as far as additional complexity is concerned. Another few methods of achieving the same goal could be: (a) instead of exchanging bits during time-critical phases one could take the approach of Rasmussen and Čapkun [69] of exchanging longer bit strings in constant stream (however, the bitstrings should reveal some information about a secret key if they are all sent – this ensures terrorist fraud resistance); (b) use a much larger number of rounds for the distance-bounding, out of which the verifier picks a small subset of arbitrarily chosen phases, thus deciding whether to authenticate the prover or not (however, this would create a very large communication complexity in order to ensure asymptotical security).

5 In-Depth: Terrorist Fraud Insights

In this chapter we return to the topic of terrorist fraud resistance. As noted in previous chapters, especially in Chapter 2, terrorist fraud is the strongest attack we consider in single-prover-single-verifier distance-bounding scenarios, and its definition is a subject of controversy. Intuitively speaking, this attack is performed by an adversary colluding with a dishonest prover. In general, the dishonest prover is assumed to interact with the adversary offline, aiding the adversary in its authentication attempt. The only restriction on the information forwarded by the dishonest prover is that the adversary should be unable to authenticate on its own.

In concurrent work to ours, Avoine et al. [4] define terrorist fraud resistance somewhat informally, requiring that the dishonest prover's aid gives the adversary "no further advantage" in later authentication attempts. It is unclear though what "further" means here, and how to define it. In ulterior work, Avoine et al. [5] give a proof of terrorist fraud resistance in their model by relying on a proof of zero-knowledge of the secret key by the adversary. However, this seems too strong a requirement, since not all information about the secret key can be used by the adversary immediately. We discuss this issue in detail in Section 2.2.2.

By contrast, our simulation-based definition in Chapter 2 is very precise and very strong. A protocol is terrorist fraud resistant if an adversary (aided by the prover) authenticates with some probability only if a simulator can (on its own) recover enough information from the adversary's state to authenticate *with the same probability*. In practice thus, the protocol must "force" the dishonest prover into giving the adversary directly-usable non-trivial information, like a long-term secret key.

Our comparative protocol analysis in Section 2.3 shows that traditional approaches towards distance bounding, as exemplified in [70] and [50], are not effective in achieving terrorist fraud resistance. In fact, it seems tricky to prove that *any* existing protocol in the literature is terrorist fraud resistant in our definition. This is due to the strength of our terrorist fraud definition. In fact, in Chapter 2 we show an attack against the Reid et al. protocol of [70], where we exploit a subtlety of our model: here, both the adversary and the simulator will be able to authenticate, but with different probabilities (the simulator will be at a disadvantage since, unlike the adversary, it cannot query the dishonest prover). We refer the reader to Section 2.3.5 for more details. This attack seems excluded by the informal definition of [4], further indicating thus that our model might be too strong. On the other hand, using the definition due to Avoine et al. seems rather difficult, as it is too informal and not generic enough.

In this chapter we explore two main research directions, which follow naturally from the definitions we have seen thus far for terrorist fraud resistance.

1. Is there a scheme that is provably terrorist fraud resistant in the model we show in Chapter 2?
2. Since the definition in Chapter 2 is very strong, and the one in [4] is too informal, can we formalise terrorist fraud resistance so that the intuition is fully captured without excluding efficient constructions?

Our results in this section (also presented in [31]) answer the first question positively, as we will see. Whereas it seems unlikely that any construction in the literature is terrorist fraud resistant in the definition of Chapter 2, we can show that a modification of traditional constructions, allowing a back door for the simulator to authenticate once the adversary does so successfully, achieves our strong notion. However, our main focus in this chapter is the second research direction, namely finding a definition that both captures the intuition behind this attack and allows for efficient protocols.

The roadmap for this section is as follows:

- We begin by describing the first scheme that achieves terrorist fraud resistance in our model. In the interest of legibility, we call the terrorist fraud resistance definition in Chapter 2 Simulation-based Terrorist Fraud resistance, in short, SimTF security.
- In our quest to accurately cover the notion of terrorist fraud resistance, we begin by noting that in SimTF security, terrorist adversaries only query dishonest provers during lazy phases, i.e., offline. In many cases in practice, this may be a reasonable model. However, in high-security scenarios (e.g. when authentication enables access across a border, or into a top-secret facility), we may assume that the dishonest prover is willing to provide online assistance. In such scenarios, we want to ensure that our model is very strong, in the sense that even powerful adversaries cannot win against schemes that are proved terrorist fraud resistant in this definition. We thus introduce the *strong simulation-based terrorist fraud model* (strSimTF), where the dishonest prover can also be queried *during time-critical phases*, though the scheduling of the messages should not be as in pure relay. This definition is still simulation based, and like in Chapter 2, the notion of terrorist fraud resistance compares the success probability of the adversary (colluding with the dishonest prover) to that of the simulator.

- Having provided a complete, strong definition for high-security scenarios, we turn to our original goal of formalizing a terrorist fraud definition which allows for efficient construction, while also capturing the intuition behind this strong attack. As a third contribution, we introduce a game-based model, which comes much closer to the intuition of Avoine et al. [4]. This model, which we call game-based terrorist fraud, in short GameTF, is different from simulation based security, in the sense that it focuses on the ulterior advantage of the adversary after interacting with the dishonest prover. We consider a protocol secure if an adversary authenticating with the prover's adversary is able to authenticate afterwards with better probability than it would have otherwise, in a regular MITM attack. This model is independent of our previous approach, but captures the intuition of terrorist fraud. Noting that the protocol due to Reid et al. was proved to be SimTF-insecure in Chapter 2, we prove that this protocol, which is considered "intuitively" terrorist fraud resistant, is indeed provably GameTF secure. This also provides a separation between the notions of SimTF and GameTF security.
- Finally, we complete the overview we attempt to give with this chapter by relating the models and showing separations and implications between the various notions. Interestingly, the strSimTF and GameTF models are independent; however, a protocol achieving strSimTF security *and* mafia fraud resistance is also GameTF secure. We also show that, though our GameTF definition resembles the informal notion in [4], it does *not* imply mafia fraud resistance (contrary to the results of [4]). The full map of the relationships is shown in Figure 28.

5.1 A Terrorist Fraud Resistant Protocol

We begin by briefly recalling the intuition behind the terrorist fraud resistant model we showed in Chapter 2.1, which we abbreviate from now on SimTF security. We then also briefly describe the intuition behind the general terrorist fraud attack we showed in Section 2.3.5, against the protocol due to Reid et al. [70]. Then we proceed to describe the first SimTF secure scheme in the literature, describing in particular why it bypasses our previous attack.

5.1.1 An Overview of SimTF Security

In Chapter 2, the communication model is defined in terms of sessions; in the mafia fraud model, we say that time-critical phases of a verifier-adversary session (in the RFID terminology used in Chapter 2, this corresponds to reader-adversary sessions) are *tainted* by an adversary-prover (resp. adversary-tag) session if the adversary relays exact communication between an honest prover and an honest verifier. Since the definition of tainted phases will be crucial in this chapter, we recall the following very useful and intuitive sketch in Chapter 2, depicting tainted phases.

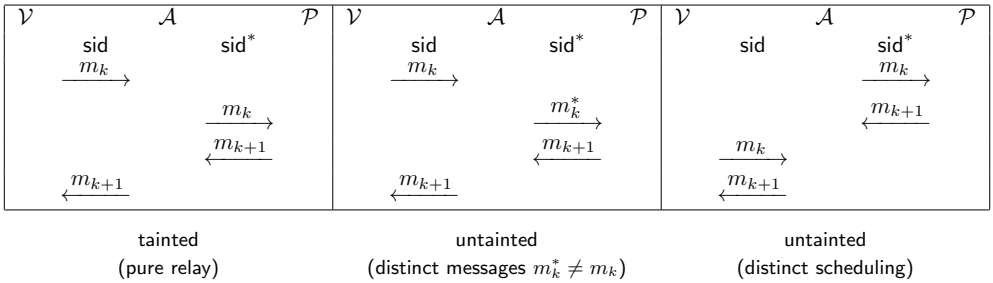

Figure 23: Mafia Fraud: tainted and untainted time-critical phases

SimTF **Security.** The definition of tainted phases naturally extends for SimTF security. We denote the dishonest prover by $\mathcal{P}'$ (an apostrophe in this chapter will always denote that a party behaves dishonestly, in accordance to our notation in Chapter 2). A SimTF adversary doesn't need to run prover-verifier sessions, and may query $\mathcal{P}'$ only during lazy phases; if he queries $\mathcal{P}'$ in time-critical phases, the phase is tainted.

Definition 5.1 (Tainted Time-Critical Phase (SimTF)) *A time-critical phase* $\Pi_{\mathsf{sid}}[k \dots k+2\ell-1] = (m_k, \dots, m_{k+2\ell-1})$ *for* $k, \ell \geq 1$ *of a verifier-adversary session* sid*, with the* k*-th message being received by* $\mathcal{A}$*, is* tainted *if there exists a session* sid′ *between* $\mathcal{A}$ *and* $\mathcal{P}'$ *such that, for some* i*,*

$$\mathit{clock}(\mathsf{sid}, k) < \mathit{clock}(\mathsf{sid}', i) < \mathit{clock}(\mathsf{sid}, k+2\ell-1).$$

SimTF security is defined in terms of a simulator: once an adversary $\mathcal{A}$ authenticates in a verifier-adversary session, its transcripts and randomness (i.e. the view $\text{view}_{\mathcal{A}}$ of $\mathcal{A}$) are passed to a simulator $\mathcal{S}$ which must authenticate, by only using $\text{view}_{\mathcal{A}}$, with at least as much probability. Thus, if the adversary requests (a part of) the secret key, this information is passed on to the simulator.

Definition 5.2 (SimTF **Security)** *Let* DB *be an authentication scheme for parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$*. Let* $\mathcal{A}$ *be a* $(t, q_{\mathcal{V}}, q'_{\mathcal{P}})$-SimTF *adversary,* $\mathcal{S}$ *be an algorithm with runtime* $t_{\mathcal{S}}$*, and* $\mathcal{P}'$ *be an algorithm with runtime* t'*. Denote*

$$\mathbf{Adv}_{\mathsf{DB}}^{\mathsf{SimTF}}(\mathcal{A}, \mathcal{S}, \mathcal{P}') = p_{\mathcal{A}} - p_{\mathcal{S}}$$

where $p_{\mathcal{A}}$ *is the probability that* $\mathcal{V}$ *accepts in one of the* $q_{\mathcal{V}}$ *verifier-adversary sessions* sid *such that at most* $T_{\max}$ *time-critical phases of* sid *are tainted, and* $p_{\mathcal{S}}$ *is the probability that, given* $\text{view}_{\mathcal{A}}$*,* $\mathcal{S}$ *authenticates to* $\mathcal{V}$ *in one of* $q_{\mathcal{V}}$ *subsequent executions.*

We discuss the intuition behind this definition. In [4], the adversary is an active party with a twofold goal: to authenticate with the prover's aid; and then to authenticate without permission. This approach excludes many attacks, as almost *any* data forwarded by $\mathcal{P}'$ *will* give $\mathcal{A}$ some advantage in future attacks. By contrast, SimTF security focuses on the information forwarded by $\mathcal{P}'$ to $\mathcal{A}$, requiring that if $\mathcal{A}$ authenticates to $\mathcal{V}$, its state contains data making future authentication attempts as successful. Excluded are only attacks where prover data is *directly* used by $\mathcal{S}$.

Attack Description (informal). In fact, the attack we showed in Section 2.3.5 against the scheme of Reid et al. exploits precisely the fact that the adversary gains an advantage to impersonate compared to the simulator, because the prover's information cannot be directly used by the simulator. In particular, in this protocol, the prover uses a pseudorandom function (PRF) in order to first compute an ephemeral response string R^0, which depends on nonces chosen by the prover and the verifier (thus they are fresh for every session). A second response string R^1 is then computed as a symmetric encryption, under a long term secret key. Thus, knowledge of either R^0 or R^1 reveals no direct information about the long term secret key, which is what provides our adversary with an advantage over the simulator.
Concretely, in our attack the dishonest prover gives the adversary one of the two responses, namely the pseudorandom R^0 in every verifier-adversary session that this adversary runs. During the time-critical phases of the protocol, the verifier always forwards a single bit c as the phase's challenge, expecting as a response a bit of the response string R^c. Thus, in our attack, the adversary always knows the correct response for phases whose challenges are $c = 0$. If the challenge is 1, then the adversary guesses the corresponding response. The overall success probability of this attack is thus roughly $\frac{3}{4}^{N_c-(T_{\max}+E_{\max})}$, where N_c denotes the number of time-critical rounds run by the protocol, $T_{\max}$ is the maximum number of rounds the adversary can taint, and $E_{\max}$ is the maximum number of rounds with erroneous responses.
Once the adversary succeeds once, the simulator receives the adversary's transcripts. Note first that the simulator has no way of knowing, for any failed verifier-adversary authentication sessions, which time-critical phases (amongst those with challenge $c = 1$) failed, and which succeeded. Also, the simulator cannot taint phases. However, even if we assume that $T_{\max} = E_{\max} = 0$, the simulator learns only roughly $\frac{N_c}{2}$ bits of the long term secret (since for the phases where $c = 1$, in the successful authentication attempt, the adversary learns the bits of both R^0 and R^1), and its only way of authenticating is to guess the response bits in every round, amounting to a total success probability of $\frac{1}{2}^{N_c}$.
This attack can be further scaled, in case we wish to give the adversary a better probability to authenticate: in this case, the prover also gives the adversary a part of R^1. Even if the simulator's probability to authenticate may increase to being non-negligible, this probability is never as large as the adversary's. Thus, the scheme is provably *not* terrorist fraud resistant.

5.1.2 The Protocol

Let PRF denote a pseudo-random function (PRF). We follow the idea of [15, 70], where the prover encrypts a long-term secret key with a value output by PRF. The prover runs PRF on a pair of nonces, one prover-chosen, the other, verifier-chosen, computing a response R^0. The second response R^1 can be seen as a one-time-pad encryption of the first response R^0 with *a* secret key *sk′*. As an important assumption, we note that we assume *sk′* to be generated by the key-generation algorithm Kg by either the reader or an honest third party; furthermore, the key is drawn at random from a distribution $\mathcal{D}$ that is computationally indistinguishable from the uniform random distribution. In fact, since $R^1 = R^0 \oplus sk'$, this will later ensure that the protocol is distance-fraud resistant, despite the fact that a dishonest tag can manipulate the pseudorandom function PRF and choose a "weak" nonce, so as to increase its success probability in distance fraud (see the recent attack of Boureanu et al. [11]).

Apart from granting distance fraud resistance, sk' also enables a simulator to authenticate if $\mathcal{P}'$ reveals both R_i^0 and R_i^1 for some phase i (thus forwarding sk'_i). However, $\mathcal{A}$ can taint rounds, while $\mathcal{S}$ cannot; to account for tainted rounds and erroneous responses, the simulator may reconstruct some bits of sk' wrongly, as the verifier will —with some probability— authenticate $\mathcal{S}$ for a wrong guess sk'' of sk'. The authentication probability depends on the Hamming weight $\#_1(\cdot)$ of the difference between sk'' and sk' (see Figure 24). Note that, as most other terrorist-fraud protocols have a security level of N_c bits, i.e. the number of time-critical rounds, this authentication method does not lower the security of our scheme, which is (up to $T_{\max}$ and $E_{\max}$) equal to the size $N_c = |sk'|$ of the secret key, thus also sufficiently large.
A crucial part of our compiler, apart from allowing the simulator to guess some bits of sk' wrongly, is that once $\mathcal{S}$ uses this guess, it does *not* need the fresh sessions responses R^0 and R^1. In particular, the verifier has a flag a, set by default to 0. However, if the simulator uses its guess of sk' to authenticate, and if this value passes the lazy phase verification, then $\mathcal{V}$ sets $a = 1$. During the time-critical rounds, if $a = 1$, the verifier accepts as responses the echoed challenges, which $\mathcal{V}$ chooses at random. Note that this does not help the dishonest prover for distance fraud, as the challenges are unpredictable.
In order to prevent mafia fraud key-learning attacks (see Chapter 4), we use the strong mafia-fraud resistance compiler of Section 4.1 and add a last lazy authentication phase, where the prover computes a pseudorandom function output on the entire transcript of the protocol. We use a different pseudorandom function than required by the rest of the protocol, which we call F, and we denote the transcript by τ_{DB}. If in the Simulator mode (i.e. if $a = 1$, then P is not verified).
We show the full protocol in Figure 24.

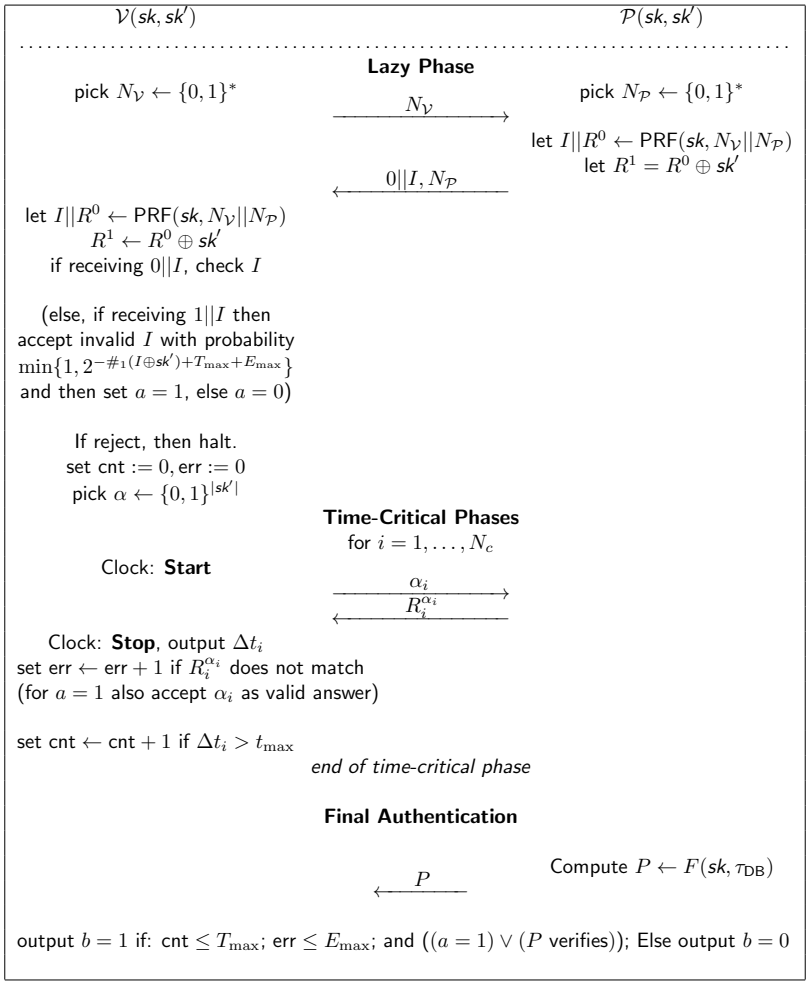

Figure 24: SimTF secure distance-bounding protocol.

5.1.3 Security

In this section, we prove SimTF security for our scheme, under the assumption that verifier-adversary sessions are executed sequentially.

Theorem 5.3 (SimTF **Security)** *Let* DB *be the distance-bounding authentication scheme in Figure 24 with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$. *For any* $(t, q_{\mathcal{V}}, q_{\mathcal{P}'})$-SimTF *adversary* $\mathcal{A}$ *against the scheme, mounting a sequential attack, there exists a* $t_{\mathcal{S}}$*-simulator* $\mathcal{S}$ *with* $t_{\mathcal{S}} = 2t + O(nq_{\mathcal{V}})$ *such that we have*

$$\mathbf{Adv}_{\mathsf{DB}}^{\mathsf{SimTF}}(\mathcal{A}, \mathcal{S}, \mathcal{P}) \leq 0.$$

Proof. We describe the simulator $\mathcal{S}$. Given $\mathsf{view}_{\mathcal{A}}$, including $\mathcal{A}$'s randomness, $\mathcal{S}$ internally runs $\mathcal{A}$ stepwise with $\mathsf{view}_{\mathcal{A}}$. Note that $\mathcal{S}$ has as many attempts as $\mathcal{A}$ to authenticate; thus $\mathcal{S}$ repeats its strategy for each verifier-adversary session sid. Namely, $\mathcal{S}$ checks if $\mathcal{A}$ sends $1||I$ and succeeds; if so, $\mathcal{S}$ sets $sk'' = I$ for sid. Else, if $\mathcal{A}$ uses $0||I$, the simulator constructs sk'' as follows: each time $\mathcal{A}$ expects α_i in the next time-critical phase, $\mathcal{S}$ branches into two executions, once sending $\alpha_i^0 = 0$ and the other time $\alpha_i^1 = 1$ to $\mathcal{A}$. It waits for $\mathcal{A}$ to answer in both branches, or query $\mathcal{P}'$ (tainting the branch). As we consider sequential executions only, there are no other options. If $\mathcal{A}$ taints or refuses one query, $\mathcal{S}$ picks sk_i'' at random; else it sets $sk_i'' = R_i^0 \oplus R_i^1$. The simulator returns to its main execution and resumes the simulation with the correct α_i. When $\mathcal{A}$ stops, $\mathcal{S}$ has predictions sk_i'' for each bit of sk_i'.
Now note that, if $\mathcal{A}$ succeeds in some sid with $0||C$, then there are four cases for each bit sk_i'' guessed by $\mathcal{S}$:

- The adversary taints the phase or refuses to answer both challenges. Then $\mathcal{S}$'s guessing strategy is good: by comparing the term $\#_1(I \oplus sk') - T_{\max} - E_{\max}$ (i.e. the number of bits $\mathcal{S}$ needs to predict) to the number of phases $\mathcal{A}$ needs to pass, we see that $\mathcal{S}$ gets a "wild card" for each of the at most $T_{\max}$ tainted phases. Thus, if $\mathcal{A}$ taints the phase in both branches, it succeeds for one round; however $\mathcal{S}$ then "gains" 1.5 bits by deducting one wild card off $T_{\max}$ and guessing a bit of sk' with probability $\frac{1}{2}$. Thus $\mathcal{S}$ has an advantage over $\mathcal{A}$. If, on the other hand, $\mathcal{A}$ taints in exactly one branch *and* it always responds correctly in the other branch (it always wins the round, unlike $\mathcal{S}$), then $\mathcal{S}$ gets half a bit from sk_i' correctly (for the untainted branch, which occurs according to α_i with probability $\frac{1}{2}$), and another half a bit from the tainted branch ($\mathcal{A}$ cannot taint another round later). Thus, $\mathcal{S}$ gets, on average, the same number of bits as is the adversary's success probability.
- Both R_i^0 and R_i^1 given by $\mathcal{A}$ are correct, so $\mathcal{A}$ wins the round. Then $sk_i'' = sk_i'$, and $\mathcal{S}$ gains one bit.
- Analogously, $sk_i'' = sk_i'$ if both replies are incorrect ($\mathcal{A}$ fails here).
- If exactly one of R_i^0 and R_i^1 is correct, then sk_i'' is certainly incorrect. But then $\mathcal{A}$ too fails the phase with probability $\frac{1}{2}$. The reasoning from the first case for $T_{\max}$ applies to $E_{\max}$.

Thus, accounting for at most $T_{\max} + E_{\max}$ tainted and erroneous phases, $\mathcal{A}$ authenticates with probability at most $2^{-\#_1(sk'' \oplus sk') + T_{\max} + E_{\max}}$. By using sk'', $\mathcal{S}$ also authenticates with the same probability. Also, if $\mathcal{S}$ reuses $sk'' = I$ for adversary executions with $1||I$, it succeeds with the same probability as $\mathcal{A}$. □

Proposition 5.4 (Mafia Fraud Resistance) *Let* DB *be the scheme in Figure 24 with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$. *For any* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-mafia-fraud adversary* $\mathcal{A}$ *against the scheme there exists a* (t', q')*-distinguisher* $\mathcal{A}'$ *against* PRF *(where* $t' = t + O(n)$ *and* $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$*) such that:*

$$\begin{aligned}
\mathbf{Adv}_{\mathsf{DB}}^{\textit{mafia}}(\mathcal{A}) \quad \leq \quad & \left(\tfrac{1}{2}\right)^{N_c - (T_{\max} + E_{\max})} + 2\binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-(|N_{\mathcal{V}}| + \lceil \frac{N_c}{2} \rceil - T_{\max} - E_{\max})} \\
& + 2\binom{q_{\mathcal{P}}}{2} \cdot 2^{-(|N_{\mathcal{P}}| + \lceil \frac{N_c}{2} \rceil - T_{\max} - E_{\max})} + 2\binom{q_{\mathcal{P}}}{2} \cdot 2^{-(|N_{\mathcal{P}}| + N_c - 1 - T_{\max} - E_{\max})} \\
& + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-(|N_{\mathcal{V}}| + N_c - 1 - T_{\max} - E_{\max})} + q_{\mathcal{V}} \cdot (\mathbf{Adv}_{\mathsf{PRF}}^{d}(\mathcal{A}') + \mathbf{Adv}_{F}^{d}(\mathcal{A}'')) \\
& + q_{\mathcal{V}} \cdot 2^{-(2 - \log_2 3) N_c + T_{\max} + E_{\max}}.
\end{aligned}$$

Proof. This proof follows the lines of the proof of mafia fraud security for the Swiss Knife protocol in Section 2.3.6, with one exception. A mafia fraud adversary against this protocol has two choices: (a) the adversary tries to guess (or learn) the key sk', thus making the verifier switch a to 1; or (b) the adversary opens a parallel adversary-prover session and forwards the authentication string I. For the latter case, the proof follows the lines of the mafia fraud resistance proof for the Swiss

Knife protocol, see Section 2.3.6. In particular, the main steps of the proof are: first the PRF output of the function PRF is replaced by truly random values (this can be done since the key used in generating the output I cannot be learned and is not used elsewhere); then (with a negligible loss of security) we assume that at most a single adversary-prover session shares nonces with the winning verifier-adversary session; and finally the probability of winning is upper-bounded. An important observation here is that if the adversary tries to use a Go-Early strategy, there is a high probability this fails, due to the final authentication lazy phase. In case (a) for each verifier-adversary session the adversary has some probability of guessing sk' or a value sufficiently close to sk' to authenticate. One way of learning sk' is to try a key-learning attack; however, this succeeds only with negligible probability, as discussed both in the proof of mafia fraud resistance for the Swiss Knife protocol and in Chapter 4. Another way is to guess the value. Now $\mathcal{A}$ chooses I with $\#_1(I \oplus sk') = m$ for some m with probability $\binom{N_c}{m}2^{-(N_c-m)}2^{-m}$. The verifier accepts C w.p. $2^{-m+T_{\max}+E_{\max}}$. Summing over all m, the overall probability is $q_{\mathcal{V}}2^{-(2-\log_2 3)N_c+T_{\max}+E_{\max}}$ (using that $(1+X)^{N_c} = \sum_{k=0}^{\infty}\binom{N_c}{k}X^k$) for all $q_{\mathcal{V}}$ sessions (note that sk' is never revealed by an honest prover). □

Proposition 5.5 (Distance Fraud Resistance) *Let* DB *be the scheme in Figure 24 with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$. *Furthermore, assume that* Kg *is run by either the reader or a third, trusted party that is* not *the tag, such that it generates keys* sk, sk' *by drawing them uniformly at random from a distribution* $\mathcal{D}$ *computationally indistinguishable from the uniform random distribution. For any* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-distance-fraud adversary* $\mathcal{A}$ *against* DB *it holds that,*

$$\mathbf{Adv}^{dist}_{\text{DB}}(\mathcal{A}) \leq q_{\mathcal{V}} \cdot \left(\frac{3}{4}\right)^{N_c - T_{\max} - E_{\max}} + \mathbf{Adv}^{dist}_{\mathcal{D}}(\mathcal{A}').$$

Proof. In this proof, we cannot replace the PRF output by (independent) random values, as in the proof above (as verbally indicated by Serge Vaudenay, the prover can manipulate the PRF instance to increase its winning probability [11]). However, we can follow the same steps as in the proof of distance-fraud resistance for the Kim et al. Swiss-Knife protocol [50], with a single modification. Namely, if the tag chooses to send $1||sk'$ as a lazy phase response (note that the distance-fraud adversary knows the second secret key sk'), then the bit a is set to 1, and the reader expects his challenges to be echoed. Since the challenges are chosen independently and uniformly at random, the probability that the tag ca commit to the correct response of every phase is $\frac{1}{2}$. However, if the tag uses $0||I$, forwarding the correct I to the reader, the proof resembles that for the Swiss-Knife protocol. The first, crucial step is replacing the value of sk by a truly random value (at the expense of a term $\mathbf{Adv}^{\text{dist}}_{\mathcal{D}}(\mathcal{A}')$); then, assuming sk is a uniform random value, it holds that $T_i^0 = T_i^1$ with probability $\frac{1}{2}$ (in which case a distance-fraud adversary wins the round), and $T_i^0 \neq T_i^1$ with probability $\frac{1}{2}$ (in which case the adversary can only guess the correct response). This adds up to the bound above. □

Proposition 5.6 (Impersonation Security) *Let* DB *be the scheme in Figure 24 with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$. *For any* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-impersonation adversary* $\mathcal{A}$ *against* DB *there exists a* (t', q')*-distinguisher* $\mathcal{A}'$ *against* PRF *(with* $t' = t + O(n)$ *and* $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$*) such that*

$$\begin{aligned}\mathbf{Adv}^{imp}_{\mathcal{ID}(\mathcal{A})} \leq\ & q_{\mathcal{V}} \cdot 2^{-|I|} + q_{\mathcal{V}} \cdot 2^{-(2-\log_2 3)N_c + T_{\max} + E_{\max}} \\ & + \mathbf{Adv}^{d}_{\text{PRF}}(\mathcal{A}') \cdot \mathbf{Adv}^{d}_{F}(\mathcal{A}') + \left(\binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}\right) \cdot 2^{-N_c}.\end{aligned}$$

Proof. The proof is similar to the proof of the scheme in [27]. We can replace the PRF output I by random values by losing a term $\left(\binom{q_{\mathcal{V}}+q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}\right) \cdot 2^{-N_c}$ security (the factor 2^{-N_c} accounts for the fact that even if the nonces are shared and thus I is identical across two different sessions, the adversary still needs to have identical challenges/responses in order to be able to provide a correct value P). Then, having replaced the PRF output by random values (as in the mafia fraud proof), $\mathcal{A}$ must now guess I (with probability $2^{-|I|}$) or send some $1||I$, such that $\mathcal{V}$ accepts. As in the mafia fraud resistance proof, the adversary is successful with probability $q_{\mathcal{V}}2^{-(2-\log_2 3)N_c+T_{\max}+E_{\max}}$. □

5.2 Flavours of Terrorist Fraud Resistance

In this section we proceed to address our second research question for this chapter, extending the formal, exact SimTF model of Chapter 2. We first argue that, whereas the intuition of terrorist fraud is that the dishonest prover only helps the adversary *offline*, it may be useful to consider dishonest provers providing *online* help to adversaries, e.g. for scenarios where authentication may yield big rewards, such as bypassing border controls, entering high-security areas, etc. In Section 5.2.1, we introduce strong simulation-based terrorist fraud resistance strSimTF, where the adversary receives online help from the prover.

Next, we argue that, whereas in Section 2.3 we show an attack against schemes like [70], this protocol *does* attain some intuitive terrorist fraud resistance, as the prover's help gives the adversary an advantage to win subsequent attempts. Thus, in Section 5.2.2 we define a game-based terrorist fraud model GameTF, which captures this intuitive degree of security, and we show that the scheme in [70] is GameTF secure.

5.2.1 strSimTF Security

Our main motivation in considering stronger simulation-based terrorist fraud resistance is exactly that terrorist fraud is a very strong attack. Thus it makes sense to give the adversary as much power as possible, and in particular, to allow as much interaction as possible between the adversary and the dishonest prover.

Our notion of Strong Simulation-based terrorist fraud resistance strSimTF enables $\mathcal{A}$ to communicate with $\mathcal{P}'$ in time-critical phases, but relaying at most $T_{\max}$ phases. Our notion of tainted phases for strSimTF security resembles that of tainted mafia fraud phases (see Section 5.1.1), but it is more restrictive, as $\mathcal{A}$ can no longer flip bits in its interaction with $\mathcal{P}'$. This restriction is necessary to prevent a trivial attack, which exploits the fact that the prover is dishonest (thus, in a sense, the adversary has non-black-box access to the prover, whereas mafia adversaries communicate with the prover as with a black box).

This trivial attack goes as follows. Consider a single time-critical phase, where $\mathcal{V}$ forwards a challenge c and expects response r_c depending on c. If the protocol is mafia fraud resistant, then r_c reveals no information about the secret key. Assuming mafia fraud resistance, the dishonest prover cannot forward the correct r_c without learning c. But, if $\mathcal{A}$ can query $\mathcal{P}'$ during time-critical rounds, it gets the challenge c, sends it to $\mathcal{P}'$, which forwards, instead of r_c, the value $1||r_c$. Now $\mathcal{A}$ removes the first bit and forwards r_c. Since $\mathcal{A}$ learns nothing about the secret key, it wins with probability 1, while $\mathcal{S}$ has no chance to win.

Our strSimTF model resembles SimTF security; however, instead of querying $\mathcal{P}'$ in at most $T_{\max}$ time-critical phases, tainting them, the strSimTF adversary can query $\mathcal{P}'$ during *all* time-critical phases, tainting them only by relaying scheduling (whether the bits are flipped or not). This adversary has more power; in fact, we show that there exist SimTF secure protocols that are strSimTF insecure (see the separation in Theorem 5.13).

Following our approach in Chapter 2 we want to exclude as few attacks as possible. In strSimTF security, the adversary may also communicate with the dishonest prover in time-critical rounds, with the restriction that the *order* of the messages is strictly not that of a relay attack. This is captured in the figure below, which shows which time-critical phases are tainted during strSimTF and which are not. Note that, as opposed to tainted phases in mafia fraud attacks, the middle example *is* in fact tainted, due to the scheduling, even though it does not correspond to a pure relay.

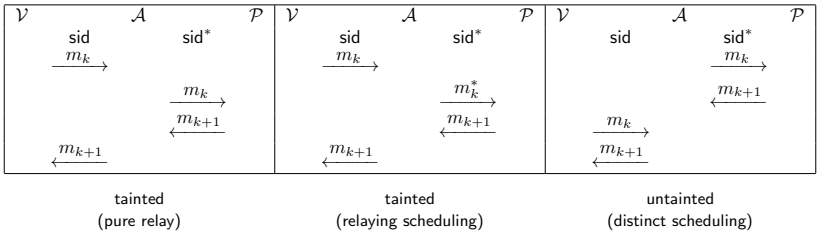

Figure 25: strSimTF security: tainted and untainted time-critical phases

Definition 5.7 (Tainted Time-Critical Phase (strSimTF)) *A time-critical phase* $\Pi_{\mathsf{sid}}[k \dots k{+}2\ell{-}1] = (m_k, \dots, m_{k+2\ell-1})$ *for* $k, \ell \geq 1$ *of a verifier-adversary session* sid*, with the* k*-th message being received by the adversary, is* tainted *if there exists an adversary-prover session* sid* *and messages* $(m_k^*, \dots, m_{k+2\ell-1}^*)$ *such that for all* $i = 0, 1, \dots, \ell - 1$ *we have:*

$$\mathit{clock}(\mathsf{sid}, k + 2i) < \mathit{clock}(\mathsf{sid}^*, k + 2i),$$
$$\textit{and} \qquad \mathit{clock}(\mathsf{sid}, k + 2i + 1) > \mathit{clock}(\mathsf{sid}^*, k + 2i + 1).$$

We use the approach of SimTF security, comparing the success probability of a prover-aided adversary to that of a simulator using only $\mathsf{view}_{\mathcal{A}}$. The definition is phrased exactly as Definition 5.2, except that tainted phases follow Definition 5.7.

We prove that the scheme in Figure 24 is also strSimTF secure. This is not a trivial statement, since not all SimTF secure protocols are strSimTF secure. This relationship is discussed in Section 5.2.3.

Proposition 5.8 *Let* DB *be the protocol in Figure 24 with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$. *For any* $(t, q_{\mathcal{V}}, q_{\mathcal{P}'})$-strSimTF *adversary* $\mathcal{A}$ *against* DB*, mounting a sequential attack, there exists a* $t_{\mathcal{S}}$*-simulator* $\mathcal{S}$ *with* $t_{\mathcal{S}} = 2t + O(nq_{\mathcal{V}})$ *such that for any* $\mathcal{P}'$ *running in time* $t_{\mathcal{P}'}$*,*

$$\mathbf{Adv}_{\mathsf{DB}}^{\mathsf{strSimTF}}(\mathcal{A}, \mathcal{S}, \mathcal{P}) \leq 0.$$

Proof. We extend the proof of Theorem 5.3 to account for time-critical queries to $\mathcal{P}'$, again for sequential executions. The simulator is modified as follows: if $\mathcal{A}$ does *not* interact with $\mathcal{P}'$ during time-critical phases, the simulator is the same. If $\mathcal{A}$ *does* query $\mathcal{P}'$, for each time-critical phase where $\mathcal{A}$ interacts with $\mathcal{P}'$, the simulator branches the execution for both challenges. If $\mathcal{A}$ refuses to forward one response or taints the phase (with relay scheduling), the simulator guesses the bit in sk'' as before.
This does not change the old proof, as the protocol transcript in the strSimTF model is virtually the same, with or without the prover's aid during time-critical phases. If the phase is *not* tainted by relaying, then either $\mathcal{A}$ queries $\mathcal{P}'$ *before* challenge α_i is sent, or $\mathcal{P}'$ responds *after* $\mathcal{A}$ has responded to $\mathcal{V}$ in this phase. The former case is equivalent to the scenario where $\mathcal{P}'$ does *not* know which challenge is sent, i.e. the SimTF scenario. In the latter case, the prover's response does not help $\mathcal{A}$, as the responses are pseudorandom and independent of each other, though it may help the simulator instead (since $\mathsf{view}_{\mathcal{A}}$ contains the correct response). □

5.2.2 GameTF **Security**

The two simulation-based notions shown thus far, i.e. SimTF security as in Chapter 2 and the strSimTF security introduced in Section 5.2.1, capture the notion that if the dishonest prover helps the adversary to win (with some probability), then the provided help allows a simulator to later win with *the same* probability. This is a very strong definition, which seems achievable only if the protocol provides the simulator with an advantage in future authentication, to compensate for tainted and erroneous adversary phases. However, in e.g. logistics or public transport, more efficient constructions are needed. Thus a less restrictive terrorist fraud definition seems more indicated here, capturing the intuition that the dishonest prover helps the adversary only insofar it restricts the adversary's later access.
Our Game-based terrorist fraud resistance GameTF follows the intuition of Avoine et al. [4]. The key difference between simulation-based security and GameTF security is that GameTF security considers attacks invalid if the attacker gains an ulterior advantage to authenticate. In particular, the latter attempt's success is not measured against the terrorist adversary's success.
The model is different: we consider a simulator-free two-step game, with two adversaries $\mathcal{A}$ and $\mathcal{A}^*$ sharing the adversarial view $\mathsf{view}_{\mathcal{A}}$, as defined in the SimTF security model.[11] Now $\mathcal{A}$ runs a strSimTF interaction with the dishonest $\mathcal{P}'$, while $\mathcal{A}^*$ runs a mafia fraud interaction with $\mathcal{V}$ in the presence of the prover (who is this time honest). Now $\mathcal{A}^*$ models the adversary *after* the prover stops helping: this adversary must authenticate in a MITM attack, using $\mathcal{A}$'s state (i.e. $\mathsf{view}_{\mathcal{A}}$). Thus, whereas $\mathcal{S}$ is passive and just uses $\mathsf{view}_{\mathcal{A}}$ to authenticate, $\mathcal{A}^*$ in GameTF runs an active mafia-fraud interaction *and* uses $\mathsf{view}_{\mathcal{A}}$ to authenticate. We say that $\mathcal{A}$ *is helpful* to $\mathcal{A}^*$ if $\mathcal{A}^*$ authenticates with better than mafia-fraud-success probability (i.e. $\mathsf{view}_{\mathcal{A}}$ shouldn't help $\mathcal{A}^*$ at all). We sketch the differences between SimTF and GameTF security in Figure 26. Also note that in SimTF security, $\mathcal{A}$ queries the dishonest prover $\mathcal{P}'$ in at most $T_{\max}$ time-critical phases; however, the GameTF adversary $\mathcal{A}$ may query $\mathcal{P}'$ in *each* time-critical round, tainting the phase if it uses relay scheduling.
Of $\mathcal{A}$ and $\mathcal{A}^*$, the former is the terrorist adversary. Its attack is invalid if there exists $\mathcal{A}^*$ such that $\mathcal{A}$ is *helpful* to $\mathcal{A}^*$ (i.e. on input $\mathsf{view}_{\mathcal{A}}$, $\mathcal{A}^*$ wins a MITM attack with higher probability than any mafia fraud adversary). Schemes are GameTF secure if every terrorist fraud adversary $\mathcal{A}$ either (i) wins with negligible probability; or (ii) there exists an adversary $\mathcal{A}^*$ to which $\mathcal{A}$ is helpful. We denote $\mathcal{A}$'s runtime by t and the number of prover-verifier, resp. verifier-adversary and adversary-prover sessions it runs by $(q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}'})$. Also, $\mathcal{A}$ interacts with $\mathcal{P}'$ with the restrictions in Definition 5.7; its success probability is denoted ϵ.
Once $\mathcal{A}$ stops and forwards $\mathsf{view}_{\mathcal{A}}$ to $\mathcal{A}^*$, this adversary runs a mafia-fraud interaction with $\mathcal{P}$ (we omit the apostrophe as $\mathcal{P}$ is now honest). W.l.o.g., let $\mathcal{A}^*$ run in time $t^* \leq 3t$ ($\mathcal{A}^*$ runs $\mathcal{A}$ at most twice internally, making the same queries as $\mathcal{A}$), and let $\mathcal{A}^*$ run at most q_{OBS} prover-verifier, $q_{\mathcal{V}}$ verifier-adversary, and resp. $q_{\mathcal{V}}$ adversary-prover sessions (since $\mathcal{A}$'s queries to $\mathcal{P}'$ deviate from protocol, we allow one adversary-prover session for each verifier-adversary session for $\mathcal{A}^*$). Let $\mathcal{A}^*$ win with probability ϵ^*. We now define *helpful* terrorist fraud adversaries.

Definition 5.9 *For an authentication scheme* DB *with parameters* $(t_{\max}, T_{\max}, E_{\max}, N_c)$*, let* $\mathcal{A}$ *be a* $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}'})$ *adversary interacting with* $\mathcal{V}$ *and* $\mathcal{P}'$ *in a* strSimTF *interaction, and let* $\mathsf{st} = \mathsf{view}_{\mathcal{A}}$ *denote its state. We say that* $\mathcal{A}$

[11] Note that any other state information is computable from $\mathsf{view}_{\mathcal{A}}$, for higher runtimes.

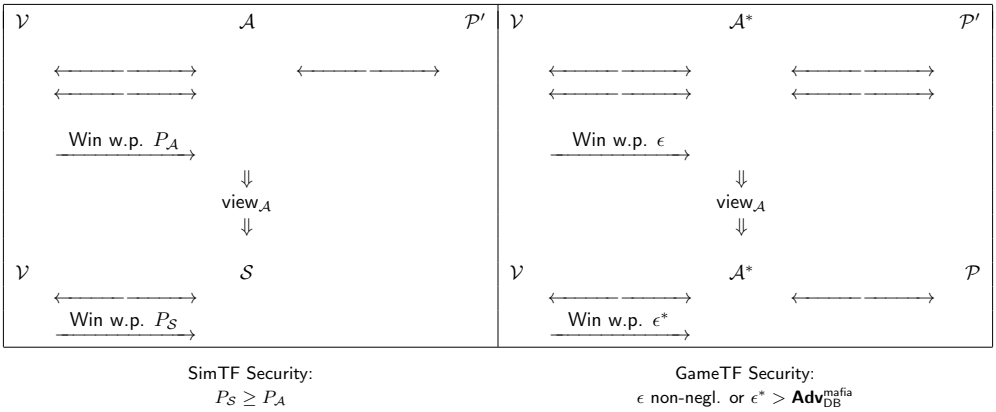

Figure 26: Simulation and game-based security models

is helpful *to an adversary* $\mathcal{A}^*$ *with input* st*, runtime at most* $3t$*, running at most* $q_{\text{OBS}}, q_{\mathcal{V}}$*, and* $q_{\mathcal{P}} = q_{\mathcal{V}}$ *sessions in a mafia-fraud interaction with* $\mathcal{V}$ *and* $\mathcal{P}$*, and winning with probability* ϵ^* *(taken over view*$_{\mathcal{A}}$ *and the coins of* $\mathcal{A}^*$*) if:*

$$\epsilon^* > \textbf{Adv}^{\textit{mafia}}_{\text{DB}},$$

where $\textbf{Adv}^{\textit{mafia}}_{\text{DB}}$ *denotes the mafia fraud resistance of* DB *for a* $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}})$*-mafia adversary.*

Now GameTF is defined as follows:

Definition 5.10 (GameTF **Security)** *Let* DB *be a distance-bounding authentication scheme with parameters* $(t_{\max}, T_{\max}, E_{\max}, \ allowbreakN_c)$*. The scheme is* $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}'}, \epsilon)$-GameTF secure *if for all* $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}'})$ *adversaries* $\mathcal{A}$ *running a* strSimTF *interaction, one of the following statements hold:*

- *The probability that* $\mathcal{A}$ *wins is upper bounded by* ϵ*;*
- *There exists an adversary* $\mathcal{A}^*$ *such that* $\mathcal{A}$ *is helpful to* $\mathcal{A}^*$ *as defined above.*

We say that a scheme DB is GameTF *secure* if it is $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}'}, \epsilon)$-GameTF secure for negligible ϵ.

A Case Study. As noted, we have already shown that one the classical, apparently terrorist fraud resistant protocol in [70] is in fact *not* SimTF secure. However, the original paper due to Reid et al. and the intuitive definition in [4] indicate that this protocol does, in fact, attain some degree of terrorist fraud resistance, which could be useful in practice. We show that schemes like [70] attain the notion of GameTF security. In particular, the attack shown in Chapter 2 is ruled out, since the adversary $\mathcal{A}^*$ gains an advantage in future authentication sessions.
In what follows, we recall the scheme of [70], as described in Chapter 2 (with the modification increasing distance-fraud security), recalling its security properties and noting that this scheme is mafia fraud resistant. We use this fact to prove GameTF security. In this scheme, $\mathcal{E}$ denotes a symmetric encryption scheme (Reid et al. suggest to use bitwise XOR in practice), while PRF denotes a pseudorandom function.

Theorem 5.11 (Reid et al. Properties) *Let* DB *be the distance-bounding authentication scheme in Figure 15 with parameters* $(t_{\max}, N_c)$*. This scheme has the following properties:*

- *It is neither impersonation resistant, distance-fraud resistant, nor terrorist fraud resistant (assuming the pseudorandomness of* PRF*).*
- *For any* $(t, q_{\mathcal{V}}, q_{\mathcal{P}}, q_{\text{OBS}})$*-mafia-fraud adversary* $\mathcal{A}$ *against the scheme there exists a* (t', q')*-distinguisher* $\mathcal{A}'$ *against* PRF *(where* $t' = t + O(n)$ *and* $q' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$*) or a* (t'', q'')*-distinguisher* $\mathcal{A}''$ *against the IND-CPA of* $\mathcal{E}$ *(where* $t'' = t + O(n)$ *and* $q'' = q_{\mathcal{V}} + q_{\mathcal{P}} + q_{\text{OBS}}$*) such that*

$$\textbf{Adv}^{\textit{mafia}}_{\text{DB}}(\mathcal{A}) \quad \leq \quad \left(\tfrac{3}{4}\right)^{N_c} + \textbf{Adv}^{d}_{\text{PRF}}(\mathcal{A}') + \textbf{Adv}^{\textit{IND-CPA}}_{\mathcal{E}}(\mathcal{A}'') + \binom{q_{\mathcal{V}} + q_{\text{OBS}}}{2} \cdot 2^{-|N_{\mathcal{V}}|} + \binom{q_{\mathcal{P}}}{2} \cdot 2^{-|N_{\mathcal{P}}|}$$

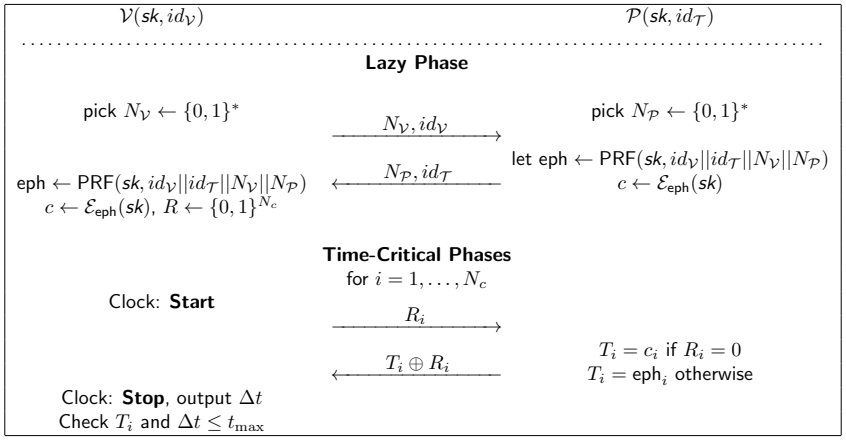

Figure 27: The Reid et al. protocol

We proceed to show that the protocol due to Reid et al. does in fact attain the game-based terrorist fraud definition we just showed.

Proposition 5.12 (GameTF **Security)** *Let* DB *be the protocol in Figure 15 with parameters* $(t_{\max}, N_c)$*. This scheme is* $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}'}, \epsilon)$-GameTF *secure, for* $\epsilon \geq \mathbf{Adv}^{mafia}_{\text{DB}}$.

Proof. Assume towards contradiction that the scheme is *not* $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}'}, \epsilon)$-GameTF resistant. Then there exists a $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}'})$ adversary $\mathcal{A}$ such that: (i) $\mathcal{A}$ wins with probability $\epsilon > \mathbf{Adv}^{\text{mafia}}_{\text{DB}}$; *and* (ii) for all $(3t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{V}})$-adversaries $\mathcal{A}^*$, initialised with view$_{\mathcal{A}}$, running a mafia fraud interaction with $\mathcal{V}$ and $\mathcal{P}$, the success probability ϵ^* of $\mathcal{A}^*$ is such that $\epsilon^* \geq \mathbf{Adv}^{\text{mafia}}_{\text{DB}}$.
We construct, for each $\mathcal{A}$ as in (i) and (ii), an $\mathcal{A}^*$ with input view$_{\mathcal{A}}$, winning in the attack above with probability $\epsilon^* \geq \epsilon$. Thus, if $\mathcal{A}$ wins with probability $\epsilon > \mathbf{Adv}^{\text{mafia}}_{\text{DB}}$ (as in (i)), our $\mathcal{A}^*$ follows the specifications of Definition 5.9 and wins with probability $\epsilon^* = \epsilon > \mathbf{Adv}^{\text{mafia}}_{\text{DB}}$ (contradicting point (ii)). Thus, an adversary $\mathcal{A}$ for which points (i) and (ii) both hold does not exist.
We describe $\mathcal{A}^*$. For each session $\mathcal{A}$ runs with $\mathcal{V}$, $\mathcal{A}^*$ runs a session with $\mathcal{V}$ and a parallel one with $\mathcal{P}$, relaying the lazy phase and running time-critical phases as follows. In each verifier-adversary session sid, $\mathcal{A}^*$ runs $\mathcal{A}$ internally, branching out in two executions as in the proof of Theorem 5.3, so that: if $\mathcal{A}$ taints a phase, so does $\mathcal{A}^*$ (both succeed with probability 1 and have 1 less phase to taint); if $\mathcal{A}$ refuses to respond to challenge $\alpha_i =: \alpha$, then $\mathcal{A}^*$ uses the Go-Early mafia strategy (see Proposition 5.4), querying $\mathcal{P}$ with challenge $\bar{\alpha} = \alpha \oplus 1$ (both $\mathcal{A}$ and $\mathcal{A}^*$ know the same response), and $\mathcal{A}^*$ guesses the response if queried with challenge α in session sid: this gives $\mathcal{A}$ and $\mathcal{A}^*$ equal winning probability; finally, if $\mathcal{A}$ forwards responses r_0^* (for a 0 challenge) and r_1^* (for a 1 challenge) for this round, $\mathcal{A}^*$ first extracts the responses $r_b = r_b^* \oplus \alpha_b$ (by cancelling the XORed challenges), and then $\mathcal{A}$ uses the Go-Early strategy, challenging $\mathcal{P}$ with $\alpha \in \{0,1\}$, and receiving R_i^{α}. Then $\mathcal{A}^*$ sets $R_i^{\bar{\alpha}} = R_i^{\alpha} \oplus r_0 \oplus r_1$; on receiving challenge $c \in \{0,1\}$ in sid, $\mathcal{A}^*$ responds with R_i^c. For the latter strategy, there are four cases:

- Both values r_0 and r_1 are correct. Then both $\mathcal{A}$ and $\mathcal{A}^*$ win with probability 1.
- Both values r_0 and r_1 are incorrect. Then $\mathcal{A}$ loses the phase, while $\mathcal{A}^*$ wins with probability 1.
- Either r_0 or r_1 is incorrect. Now $\mathcal{A}$ wins the round with probability $\frac{1}{2}$. As $\mathcal{A}^*$ runs the Go-Early strategy for challenge $\alpha \in \{0,1\}$, it knows the correct R_i^{α}, but the wrong $R_i^{\bar{\alpha}}$ (as $r_0 \oplus r_1$ is incorrect), and wins the phase with probability $\frac{1}{2}$. If they answer wrongly, both adversaries subtract 1 from $E_{\max}$.

Thus, $\mathcal{A}^*$ wins with at least as high probability as $\mathcal{A}$ in each time-critical phase, i.e. the success probability ϵ^* of $\mathcal{A}^*$ equals the success probability of $\mathcal{A}$, i.e. ϵ. It is easy to see that the parameters of $\mathcal{A}^*$ are as required. Now if there exists an adversary $\mathcal{A}$ with $\epsilon > \mathbf{Adv}^{\text{mafia}}_{\text{DB}}$, then $\mathcal{A}^*$ succeeds in its mafia-fraud interaction with probability $\epsilon^* > \mathbf{Adv}^{\text{mafia}}_{\text{DB}}$. Thus, $\mathcal{A}$ is helpful to $\mathcal{A}^*$, contradicting our assumption. □

Since the scheme is mafia fraud resistant for appropriate parameters, it is also GameTF secure.

5.2.3 Relating the Notions

We now relate SimTF, strSimTF, and GameTF security, and mafia fraud resistance, including the result that mafia and SimTF security are independent, as shown in the first chapter of this thesis.

Theorem 5.13 (Relations between Notions) SimTF*,* strSimTF*, and* GameTF *security, and mafia fraud resistance are related as in Figure 28. Arrows between notions indicate that security against one notion implies security against the other.*

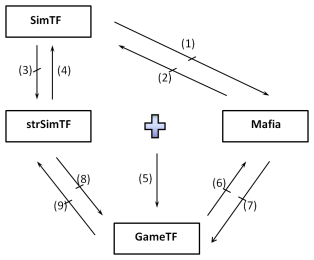

Figure 28: Relationships between the notions. Arrows indicate implications, while hatched arrows indicate separations. The "+" sign indicates composition of the properties

Proof. Relations (1) and (2) follow from Chapter 2, and (3) follows trivially from the strSimTF definition.
For relation (4), we show a separation based on our SimTF and strSimTF secure scheme in Figure 24. However, we now run $2N_c$ time-critical rounds: during odd-indexed phases $2i-1$, $\mathcal{V}$ sends the even-indexed challenges for phases $2i$, masking them with pseudorandom bits. For odd-indexed rounds, the prover must echo the challenge; for even-indexed rounds, the protocol runs as before. Thus, intuitively, the strSimTF adversary, which can communicate with the prover during time-critical rounds, (i) learns the responses for the even-indexed rounds in advance (by querying the prover for the challenges he has learned), and (ii) passes the odd-indexed rounds by simply echoing the responses.
More formally we modify scheme DB in Figure 24 to DB* as follows: firstly, PRF now outputs $I||V||R^0||R^1 \leftarrow \mathsf{PRF}(\mathit{sk}, N_{\mathcal{V}}||N_{\mathcal{P}})$, s.t. the size of V (in bits) is N_c and all the other strings are as before. Secondly, after choosing $\alpha = [\alpha_1, \ldots, \alpha_{N_c}]$, the verifier sets, for $i \in \{1, \ldots, N_c\}$: $c_{2i} = \alpha_i$ and $c_{2i-1} = \alpha_i \oplus V_i$. The scheme DB* has $2N_c$ time-critical rounds as follows: (i) time-critical phases $2i$ (i.e. even-indexed phases) are run as in Figure 24, for challenges c_{2i} and responses $R_i^{c_{2i}}$; (ii) in time-critical phases $2i-1$ (i.e. odd-indexed ones), $\mathcal{V}$ sends c_{2i-1} and expects c_{2i-1} in return. Finally, if $a = 1$ (i.e. $\mathcal{V}$ accepts $1||C$ instead of $0||C$), the verifier accepts echoed responses to all the phases. Now $\mathcal{ID}^*$ has the same properties as DB.
Though distance fraud adversaries now predict even-indexed challenges, they must still guess the odd-indexed responses. Mafia fraud adversaries trivially echo odd-indexed responses, but learn nothing about the encrypted even-indexed challenges. Finally, the SimTF security proof still stands. Firstly SimTF-adversaries and simulators trivially pass odd-indexed rounds (echoing the challenges). The simulator acts in the exact, same way, but only for *even-indexed* phases. The adversary has no advantage to win, as it may only query $\mathcal{P}'$ for $T_{\max}$ phases (note that $T_{\max}$ is unchanged), tainting them. If the adversary taints an even-indexed phase, the proof carries over. Tainting an odd-indexed phase is at most equivalent to tainting the next even-indexed phase. Thus the probabilities for both $\mathcal{A}$ and $\mathcal{S}$ remain the same.
We can show a simple strSimTF attack: $\mathcal{A}$ first forwards I and C to $\mathcal{V}$. During odd phases $2i-1$, $\mathcal{A}$ forwards $c_{2i-1} = \alpha_i \oplus V_i$ to $\mathcal{P}'$ *and* echoes this value back to $\mathcal{V}$. Since it echoes the response in time, $\mathcal{A}$ wins phase $2i-1$. At the end of the phase, $\mathcal{P}'$ recovers the value α_i and sends $R_i^{\alpha_i}$ to $\mathcal{A}$. Now $\mathcal{A}$ can use $R_i^{\alpha_i}$ in phase $2i$, once $\mathcal{V}$ sends the challenge α_i. The value $R_i^{\alpha_i}$ reveals no information about sk', thus the simulator cannot use it. Now $\mathcal{A}$ wins with probability 1, as opposed to the simulator, who succeeds with negligible probability.
For (5), let DB be a mafia-fraud and strSimTF secure authentication scheme. Suppose towards contradiction, that DB is *not* GameTF resistant. As in the proof of Proposition 5.12, we assume that there exists a $(t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{P}'})$-adversary $\mathcal{A}$ interacting in a strSimTF way such that: (i) $\mathcal{A}$ wins with non-negligible probability ϵ; (ii) all $(3t, q_{\text{OBS}}, q_{\mathcal{V}}, q_{\mathcal{V}})$ adversaries $\mathcal{A}^*$ using $\text{view}_{\mathcal{A}}$ in a mafia fraud interaction wins with probability at most $\mathbf{Adv}_{\text{DB}}^{\text{mafia}}$.
By strSimTF security, for adversary $\mathcal{A}$ there exists a simulator $\mathcal{S}$, which, given $\text{view}_{\mathcal{A}}$, wins with probability $p_{\mathcal{S}} \geq \epsilon$. Let now $\mathcal{A}^*$ take as input $\text{view}_{\mathcal{A}}$ and run $\mathcal{S}$ as a black box, winning with probability $p_{\mathcal{S}} \geq \epsilon$. This $\mathcal{A}^*$ runs a mafia fraud interaction, since $\mathcal{S}$ runs $\mathcal{A}$ internally, without interacting with $\mathcal{P}'$. Following point (ii) above, $\mathcal{A}^*$ must win with probability

at most $\mathbf{Adv}^{\text{mafia}}_{\text{DB}}$, and thus $\epsilon \leq p_{\mathcal{S}} \leq \mathbf{Adv}^{\text{mafia}}_{\text{DB}}$. Then $\mathbf{Adv}^{\text{mafia}}_{\text{DB}}$ is non-negligible, contradicting the assumption that DB is mafia-fraud resistant.

For (6) we use the Hancke-Kuhn protocol see Section2.3.2 and the original paper [42]. The mafia fraud resistance of this scheme is the same as for the Reid et al. protocol, see Chapter 2; however, a GameTF adversary can query $\mathcal{P}'$ for $R^0||R^1$ in some session sid, giving no help for future authentication. Similarly, mafia fraud security $\nrightarrow$ strSimTF.

For (7) we use a trick from Chapter 2 to modify the protocol in Figure 15 such that, if the adversary prepends a 1 bit to its nonce, the verifier expects the bit conjugate of the response. This allows the adversary to break mafia fraud resistance by flipping bits during time-critical rounds; however, a GameTF adversary cannot do this, as relay scheduling flips the phase (whether the bits are flipped or not). More formally, we modify the protocol in Fig. 15 such that in the lazy phase, an honest $\mathcal{P}$ sends $0||N_{\mathcal{P}}$; however, if $\mathcal{V}$ receives $1||N_{\mathcal{P}}$, it expects in each time-critical round, the value $R_i^{\alpha_i} \oplus 1$ instead of $R_i^{\alpha_i}$. Now a mafia fraud adversary $\mathcal{A}$ forwards $N_{\mathcal{V}}$ and flips the first bit of $0||N_{\mathcal{P}}$ such that $\mathcal{V}$ receives $1||N_{\mathcal{P}}$. For each time-critical phase, this $\mathcal{A}$ forwards the challenge α_i from $\mathcal{V}$ to $\mathcal{P}$ and receives $R_i^{\alpha_i}$. Then $\mathcal{A}$ flips this value (thus the phase is untainted) and responds correctly to $\mathcal{V}$. However, a GameTFadversary cannot use this trick, as relay scheduling taints the phase (even if the bits are flipped).

For (8), we use the same trick. In strSimTF security, the simulator must win with the same probability as an adversary. The helpfulness of GameTF adversaries depends though on mafia fraud resistance. If mafia fraud adversaries authenticate easily, any adversary is unhelpful, even one for which there exists a simulator as in strSimTF security. We modify the scheme in Figure 24 as in point (7). Formally, an honest $\mathcal{P}$ sends $0||N_{\mathcal{P}}$; however, if $\mathcal{V}$ receives $1||N_{\mathcal{P}}$, it expects in each time-critical round, the value $R_i^{\alpha_i} \oplus 1$ instead of $R_i^{\alpha_i}$. This scheme is still strSimTF secure (the proof still holds). However, $\mathbf{Adv}^{\text{mafia}}_i d = 1$, since we can construct an adversary against mafia fraud that always succeeds. This adversary $\mathcal{A}$ forwards $N_{\mathcal{V}}$ and flips the first bit of $0||N_{\mathcal{P}}$ such that $\mathcal{V}$ receives $1||N_{\mathcal{P}}$. For each time-critical phase, $\mathcal{A}$ forwards the challenge α_i from $\mathcal{V}$ to $\mathcal{P}$ and receives $R_i^{\alpha_i}$. Then $\mathcal{A}$ flips this value (thus the phase is untainted) and responds correctly to $\mathcal{V}$.

However, the scheme is GameTF insecure: an adversary $\mathcal{A}$ receiving sk' from $\mathcal{P}'$: (i) wins with probability 1; (ii) all adversaries $\mathcal{A}^*$ with input $\text{view}_{\mathcal{A}}$, win with probability at most $1 = \mathbf{Adv}^{\text{mafia}}_{\text{DB}}$.

The basic idea for (9) is to modify the scheme of Reid et al. in Figure 15 as follows: during the lazy phase, a dishonest prover (or an adversary) can send a cheating tuple of nonces and an authentication string: if the authentication string verifies, the verifier sets the challenges to the authentication string (instead of choosing them at random). This scheme is not strSimTF secure, since the scheme allows a cheating prover to know the challenges in advance and to send only the appropriate responses to the adversary, who thus learns no information about the long-term secret. However, the scheme *is* GameTF secure: if the adversary does *not* use the cheat, the proof is as before; however, if the adversary uses the cheat, the second adversary $\mathcal{A}^*$ can reuse it and then mount a Go-Early MITM strategy (which the simulator cannot do in the strSimTF scenario).

More formally, we change the scheme in Figure 15 as follows: an honest $\mathcal{P}$ always follows the protocol. But, if $\mathcal{V}$ receives in the lazy phase the value: Cheat, $N^*_{\mathcal{V}}||N^*_{\mathcal{P}}||V^*, I, N_{\mathcal{P}}, 0||C$ he checks that $V^* = [\text{PRF}(sk, N^*_{\mathcal{V}}||N^*_{\mathcal{P}})]_1^{N_c}$ (i.e. the first N_c bits of PRF's output is V^*). If this is false, $\mathcal{V}$ rejects; else, it computes $I, R^0,$ and R^1 for $N_{\mathcal{V}}$ and $N_{\mathcal{P}}$ (i.e. the fresh, not the cheating nonces), and *it sets* the challenge vector to $\alpha = V^*$.

The protocol is *not* strSimTF secure: $\mathcal{P}'$ can cheat with a triplet $N^*_{\mathcal{V}}||N^*_{\mathcal{P}}||V^*$ and learn the challenges; then it forwards the correct responses to the adversary. Thus $\mathcal{A}$ wins with probability 1. However, even if $\mathcal{S}$ knows the challenges, it cannot give the responses. However, the scheme *is* GameTF secure. The bound $\mathbf{Adv}^{\text{mafia}}_{\text{DB}}$ is unchanged (honest provers do not cheat). Assume towards contradiction that there exists a GameTF adversary $\mathcal{A}$ (i) winning with non-negligible probability; (ii) being unhelpful to all $\mathcal{A}^*$ as in Definition 5.9. We extend the adversary $\mathcal{A}^*$ in the proof of Proposition 5.12 as follows: if $\mathcal{A}$ does *not* cheat, $\mathcal{A}^*$ is as before; else, if $\mathcal{A}$ cheats, $\mathcal{A}^*$ reuses the same $N_{\mathcal{V}}{}^*||N_{\mathcal{P}}{}^*||V^*$, and uses the Go-Early strategy to query $\mathcal{P}$ with challenges V^*. Now $\mathcal{A}^*$'s wins with the same (non-negligible) probability as $\mathcal{A}$, contradicting point (ii). □

5.3 Discussion: Which Model to Use

Simulation-based models formalise terrorist fraud resistance in a very strong way, allowing attacks the prover to help the adversary as long as the gained help cannot be used by a simulator given the adversary's view only. The SimTF notion of Chapter 2 can be extended to allow online access to the prover, capturing the attack more realistically. Though strong, both SimTF and strSimTF security *can* be achieved, e.g. by our scheme. However, simulation-based security is too strong for resource-constrained devices, as it does not enable efficient protocols. In such scenarios, our game-based GameTF model is more appropriate, capturing the intuition of terrorist fraud resistance, but enabling more efficient schemes e.g. [15].

6 Location Privacy in Distance-Bounding Protocols

So far, this thesis has mostly covered the direct security of distance-bounding authentication, which is defined in terms of authentication completeness, as well as the following four properties: (i) mafia fraud resistance; (ii) terrorist fraud resistance; (iii) distance fraud resistance; and (iv) impersonation security. Furthermore, in Chapter 3 we discussed distance-bounding in the presence of key updates, where we further consider (v) long-term denial-of-service (DoS) resistance, called availability, and (vi) privacy in authentication. Notions (i) to (iv) concerns mostly security, i.e. protocols are secure if they prevent impersonation by an adversary (subject to some inherent communication-model restrictions). By contrast, availability is a requirement of achieving privacy, but also a necessary extension of the notion of completeness in the scenario of key updates.

In this chapter, we look at distance-bounding from the point of view of *location* privacy, with particular focus on the well-known construction of Rasmussen and Čapkun [69]. The results in this chapter are joint work with Katerina Mitrokotsa and Serge Vaudenay. We note that in the context of authentication and for key updates in distance-bounding, the notion of *privacy* refers in general to privacy of the identity. However, another interesting aspect of privacy at large is location privacy, where a prover does not wish to leak information about its location. In the context of distance-bounding, this is additionally tricky to achieve, since the very purpose of distance-bounding is to leak some particular information about the location, i.e. whether a prover is within proximity of a verifier or not. In this chapter we investigate whether it is possible to design distance-bounding protocols which reveal *only* this one aspect of the prover's location, i.e. proximity to a verifier.

Location privacy in distance bounding was introduced by Rasmussen and Čapkun in [69], where they noted that distance-bounding protocols may leak further location-related information than just prover-to-verifier proximity. This leakage follows from message-receival times in honest executions. To combat this, Rasmussen and Čapkun [69] proposed a privacy-preserving distance-bounding protocol (which we here call the RČ protocol). This construction was already shown to be susceptible to a non-polynomial dictionary attack which may reveal the prover and verifier locations [3]. In this chapter, we also show a mafia fraud attack against this protocol. Moreover, though Rasmussen and Čapkun intuitive *claim* that their construction is location private, the notion of location privacy has never been *formally defined* in the literature in the context of distance-bounding.

Another point which is raised in this chapter is the separation of distance-bounding from authentication, particularly addressing an idea in [69], where the authors claim their protocol achieves location-private distance bounding, but not secure authentication. Note that in the context of distance bounding, mafia, terrorist, and impersonation security all address aspects of authentication security: thus, the question here is whether distance bounding can be used as a synonym for distance fraud preservation. Since in distance fraud, the adversary is the dishonest prover, which is in possession of the secret, it follows that distance fraud secure protocols can be achieved without including secret keys, by simply having provers echo challenges from the verifier. By contrast, Rasmussen and Čapkun's protocol uses expensive primitives, including an encryption and a signature scheme. With this in mind, we show that it is easy to change the RČ protocol in order to achieve mafia fraud security and thus obtain a basic distance-bounding protocol.

Finally, a more technical contribution of this chapter, apart from formalizing location privacy, is to note that our model in Chapter 2 covers only round-based protocols, i.e. protocols where the prover and verifier interact in interleaved rounds, in a turn-based manner. By contrast, in this chapter we extend our distance-bounding framework to protocols like the RČ scheme, which feature simultaneous, continuous bit-streams sent between the prover and the verifier.

Thus, our contributions in this chapter —mostly work in submission, co-authored and jointly achieved with Serge Vaudenay and Katerina Mitrokotsa [57, 58]— could be summarised and stated as follows:

- We begin by extending our model in Chapter 2 to allow for simultaneous transmissions of messages, thus enabling us to cover the RČ protocol [69]. We capture the communication by means of two channels: a so-called *timeless* channel, where the clocks are not used, and a *timed* channel, where the time of flight of messages is measured in a bit-by-bit fashion. Both channels are duplex channels, and the adversary may use them to interact with either the prover $\mathcal{P}$ or the verifier $\mathcal{V}$ (not with both in the same session, however), or he may eavesdrop on the honest communication between $\mathcal{P}$ and $\mathcal{V}$.
- We then define a classical left-or-right indistinguishability game for location privacy in distance-bounding protocols. In this game, the adversary knows its distance to the verifier $\mathcal{V}$ and can create provers $\mathcal{P}$ at arbitrary distances from itself and $\mathcal{V}$. Our communication model is the same as the one proposed in Chapter 2, but with the modified interaction model that allows for simultaneous transmissions, via the timeless and timed channels as stipulated above. We also require that *all* the parties (not just the verifier) are associated with clocks (not necessarily synchronised[12]).

[12] These clocks, however, should measure time in a consistent way; in other words, there must exist a universal time-keeping system – which we call a time-server – such that the time measured by every party can be transformed in a consistent way in the time units of this time server.

- For our location privacy game, we consider two main adversarial classes: omniscient and limited adversaries. Omniscient adversaries capture an adversary that can measure the signal strength of transmitted messages; such adversaries are also *distributed*, in the sense that they consist of two (or more) adversaries $\mathcal{A}_1, \mathcal{A}_2$, placed at arbitrary, adversarially-chosen distances, from each other and from the verifier. This captures the notion of message triangulation. Thus, omniscient adversaries are aware, for all transmissions along the timed channel, when the message is sent and when it arrives at both the colluding adversaries. Unsurprisingly, no location privacy is feasible for omniscient adversaries. Limited adversaries, on the other hand, are only aware of the time at which they receive messages from other participants.
- Finally, we show that achieving location privacy with respect to limited adversaries is impossible in an information-theoretical sense, for protocols with a beginning or a termination, i.e. protocols where the parties do not ensure that (possibly bogus) communication is *always* exchanged in a manner indistinguishable from the protocol run itself, thus masking the beginning and end of the protocol. We prove that location privacy against limited adversaries may only be achieved if both the prover and the verifier introduce exponential delays between receiving and sending messages, and we give a lower bound for this delay. Concretely, for a security level of h bits and a maximum allowed distance of $t_{\max}$, the delay has to be at least $t_{\max}2^{h+1}$ bits. Thus, we show that location privacy can be achieved in practice, but only for very large delays, even for limited adversaries. Thanks to the high speed of light, $t_{\max}$ may still be reasonable so that we can offer practical delays (even though they are exponential).
- We finally show how to apply our results to the RČ protocol. In this context, we first review the scheme and show a mafia fraud attack against it, enabling an adversary to authenticate to the verifier, after first eavesdropping on an honest session between the legitimate prover and the legitimate verifier. The adversary in particular will try to replay, during its authentication attempt, the same nonce as the one used by the prover in the eavesdropped session. This attack can be run in a mafia fraud setting, to relay the credential of a far away prover to the verifier, which is what distance bounding was meant to avoid. The key vulnerability here is that the prover's and the verifier's messages during the initialisation phase are independent of each other, and can thus be replayed. The success of the attack depends on the probability that the adversary guesses the length offsets of the challenge, resp. the response, in the adversary's and resp. the prover's attempts to pass the protocol. Guessing the time offset depends on an adversary's ability to guess the location of the prover. We propose a new protocol based on the RČ protocol, but addressing this flaw, and then apply our location privacy results to this improved protocol.

We remark that large portions of this chapter appear almost verbatim in our submissions, jointly conceived and submitted with Katerina Mitrokotsa and Serge Vaudenay.

6.1 Model Extensions: Duplex Channels

6.1.1 Communication Model

In Chapter 2 we've considered a distance-bounding scenario with a single prover and a single verifier. In this chapter we consider multiple provers. Concretely, there is a single verifier $\mathcal{V}$, but many provers $\mathcal{P}_1, \ldots, \mathcal{P}_n$, such that $\mathcal{V}$ and $\mathcal{P}_i$ for every i share a secret key K_i output by a key generation algorithm Kg. The key K_i of each prover $\mathcal{P}_i$, and the identifier of $\mathcal{P}_i$ (in practice this identifier could be a sequence number) form an entry of a database $\mathcal{D}$ contained by the verifier $\mathcal{V}$. We also assume that when it is initialised, the verifier $\mathcal{V}$ is also equipped with an upper bound on the maximum allowed communication time (or time distance) $t_{\max}$ between itself and the prover.
The communication model we used in Chapter 2 is round-based. However, e.g. the RČ distance-bounding protocol [69] is *not* round-based. Therefore, we consider a more generalised model, where the two parties $\mathcal{P}$ and $\mathcal{V}$ interact with no round-based restriction, via *two* types of channels: a *timeless* and a *timed* channel. Parties $\mathcal{P}$ and $\mathcal{V}$ may send messages m along each of the two channels (i.e., they are duplex channels). In order to make the model more realistic we consider the transmissions along the *timed* channel to be bit-by-bit (this is so that we can model time leakage accurately, providing for the fact that receiving parties and adversaries receive bits of the message at slightly different times, depending on the packet length). In this paper, we consider a passive adversary who can only eavesdrop communication but not interfere with it. Actually, we show that we can always have a polynomially bounded passive adversary breaking location privacy.
The *timed* channel, is associated with the global TS, such that each bit of an input message m will be associated with a time ts at which the sending party has *sent* the bit. The corresponding output bit of message m is associated with a time tr, which is the time at which the receiving party has *received* the bit (note that these notations are made on the assumption of the global time server). The bit-by-bit treatment of the transmission time is compulsory, as in practice, each bit of the message is transmitted sequentially or in smaller packets. Thus, we associate a message m with an $|m|$-dimensional vector of sending times $\bar{\text{ts}}$ and an $|m|$-dimensional vector of transmission times $\bar{\text{tr}}$. We also require that the values in $\bar{\text{ts}}$ and those

in $\bar{\mathsf{tr}}$ are monotone non-decreasing, i.e. for any message m and any $1 \leq i < j \leq m$, it holds that $\mathsf{ts}_i \leq \mathsf{ts}_j$ and $\mathsf{tr}_i \leq \mathsf{tr}_j$. Furthermore, if we consider the communication between two parties A and B and that a message m is sent from the party A to the party B at time $\bar{\mathsf{ts}}$ then the reception time $\bar{\mathsf{tr}}$ of the message m at the party B will satisfy the following equation for every $i = \{1, \ldots, |m|\}$:

$$\mathsf{tr}_i = \mathsf{ts}_i + t_{AB}.$$

where t_{AB} denotes the *time distance* (mathematically speaking) between the parties A and B. More precisely, t_{AB} denotes the time (measured in time units TU) that every bit of a message m takes to travel between A and B.
Moreover, if the message m leaks off this channel to an adversary $\mathcal{A}$, each bit of the leaked message is associated with an $|m|$-dimensional timestamp $\bar{\mathsf{tr}}_{\mathcal{A}}$. Note that this information alone may not suffice to learn the *sending* time of the message, as the adversary does not necessarily know the distance between it and the sending party.
Both channels allow the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ to interact concurrently, i.e. it is possible that both the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ transmit at the same time across the duplex channel. This is indeed the case for the RČ protocol [69]. We first attach a formal definition of the two channels below.
In view of [72, 65], we consider that frequency hopping is not an effective countermeasure against eavesdropping adversaries.
We now define communication in distance-bounding protocols as being *slow* (or lazy) if it takes place on the timeless communication channel and *fast* (or time-critical) if it takes place on the timed communication channel. Note that it is possible to alternate fast and slow communication arbitrarily. Furthermore, this approach is perfectly in-tune with the similar communication model of Chapter 2, where we would now be able to use the timed and timeless channels in a round-based, turn-based manner to carry over the already-presented definitions for round-based protocols.
A direct consequence of considering continuous bitstream transmissions, which is no longer round-based, the parameter N_c is no longer directly relevant. However, we note that this parameter can translate in a number of bits of security, corresponding to the number of bits counted by the verifier's clock between the beginning of the transmission and the end of it. Since the focus of this chapter is not general distance bounding, but rather location privacy, we simplify our definition for distance-bounding authentication protocols by removing the timing parameters N_c, $T_{\max}$, and $E_{\max}$.

Definition 6.1 *We say that* $\mathsf{DB} = (\mathcal{V}, \mathcal{P}, \mathsf{Kg})$ *is a distance-bounding protocol with parameters* $(t_{\max}, \epsilon)$ *where* $t_{\max}$ *denotes the upper bound on transmission time in the fast phase and* ϵ *denotes the tolerance level for honest* $\mathcal{P}$*-*$\mathcal{V}$ *authentication failures if:*

KEY GENERATION. Kg *generates a secret key* $K \leftarrow \mathsf{Kg}(1^\ell)$ *for any* $\ell \in \mathbb{N}$.

AUTHENTICATION. *The joint execution of the prover and verifier algorithms* $\mathcal{V}$ *and* $\mathcal{P}$ *for parameters* $(t_{\max}, \epsilon)$ *ends with a verifier-generated distance bounding authentication bit* $b \in \{0, 1\}$.

We require ϵ*-completeness, i.e., the interaction of an honest prover* $\mathcal{P}$ *and an honest, fixed verifier* $\mathcal{V}$ *for parameters* $(t_{\max}, \epsilon)$ *is accepted by the verifier with probability at least* $1 - \epsilon$ *if* $t_{\mathcal{VP}} \leq t_{\max}$.

6.2 Adversarial Models

The goal of adversaries in our framework is to break location privacy as defined below. In this section, we first show how adversaries interact with the communication channels and with the honest parties during an attack. We then define two adversarial classes depending on the strength of the adversary. Finally we show the location privacy game.
We consider adversaries $\mathcal{A}$ that interact with the distance-bounding system as follows: (1) $\mathcal{A}$ may eavesdrop on the communication (across both the *timed* and the *timeless* channel) of an honest prover $\mathcal{P}$ and an honest verifier $\mathcal{V}$; and (2) $\mathcal{A}$ may interact with honest provers in prover-adversary sessions and with honest verifiers in adversary-verifier sessions. As in our framework in Chapter 2, sessions have session id's sid (practically consisting of the transcript of that session). Eavesdropping yields information to the adversary as described in Definition 6.1.
In this chapter we describe two types of adversaries. The stronger adversaries are called *omniscient*: for the timed channel transmissions, these adversaries also learn, for every bit of the message m sent by a prover, the sending time ts. This captures an adversary's ability to gauge the signal strength for a particular transmission. The adversary is able to tell how far the sender is, thus learning in particular also the sending time. Furthermore, omniscient adversaries are able to triangulate signals: we represent them as collusions of two adversaries, which both learn location data during the attack. We describe the adversarial classes below.

LIMITED ADVERSARIES. These adversaries may eavesdrop on honest prover-verifier sessions or communicate with provers and verifiers in prover-adversary and respectively adversary-verifier sessions. On eavesdropping the timed channel in honest prover-verifier sessions, these adversaries learn the transmitted message m and the bit-by-bit time the

message is received at, $\bar{\mathsf{tr}}_{\mathcal{A}} = \bar{\mathsf{ts}} + \bar{t}_{P\mathcal{A}}$, where P is the party that sent the message m and $\bar{t}_{P\mathcal{A}}$ is an $|m|$-dimensional vector with entries equalling the time distance $t_{P\mathcal{A}}$ between P and the adversary $\mathcal{A}$. Note that, as the adversary knows $t_{\mathcal{A}\mathcal{V}}$, it also learns the sending times $\bar{\mathsf{ts}}$ at which the verifier sends its messages. We should note here that the adversary $\mathcal{A}$ is able to choose its location.

OMNISCIENT ADVERSARIES. These adversaries can also eavesdrop on honest prover-verifier sessions or communicate with provers and verifiers as above. On eavesdropping on the timed channel during an honest prover-verifier session, strong adversaries learn the message m, the bit-by-bit time the message is received, $\bar{\mathsf{tr}}_{\mathcal{A}} = \bar{\mathsf{ts}} + \bar{t}_{P\mathcal{A}}$, and the bit-by-bit sending time $\bar{\mathsf{ts}}$. Thus, strong adversaries can trivially learn the distance between them and the party P that sent the message.

To justify that an omniscient adversary can also learn the sending time of messages, we could model this by distributed, *limited* adversaries, i.e. $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. The composite adversary $\mathcal{A}$ chooses the *locations* of $\mathcal{A}_1$ and $\mathcal{A}_2$ and can do triangulation of signals. This definition also extends to a *moving* adversary (i.e. an adversary that is able to change its location).

We consider only polynomial adversaries, (i.e. having polynomial run-time and running polynomially many sessions with the provers and verifier). The adversary's goal is to break the location privacy of the distance-bounding protocol, which we define by means of a left-or-right indistinguishability game as described below.

PHASE 1. In this phase, a limited adversary is given the security parameter (in unary) 1^λ. The adversary may now initialise provers $\mathcal{P}_i$ and the verifier $\mathcal{V}$ at arbitrary locations with respect to itself and the verifier, and may interact arbitrarily with the provers and the verifier. At the end of this phase, the adversary outputs two indices i, j such that $t_{\mathcal{P}_i\mathcal{V}}$ and $t_{\mathcal{P}_j\mathcal{V}}$ are both smaller than the threshold $t_{\max}$; which are forwarded to a challenger.

PHASE 2. The challenger checks that the two provers are both within the maximum distance $t_{\max}$, then closes all sessions that are open for these provers. Finally, the challenger flips a bit b and assigns the handle $\mathcal{P}_{\mathsf{Chal}}$ as follows: $\mathcal{P}_{\mathsf{Chal}} = \mathcal{P}_i$ if $b = 0$ and $\mathcal{P}_{\mathsf{Chal}} = \mathcal{P}_j$ if $b = 1$.

PHASE 3. Finally, by interacting with the challenge prover $\mathcal{P}_{\mathsf{Chal}}$, as well as all other provers with the exception of $\mathcal{P}_i$ and $\mathcal{P}_j$, the adversary must produce a decision bit d. Let $\mathsf{Exp}^{\mathsf{LocPriv}}_{\mathsf{DB}}(\mathcal{A}, 1^\lambda)$ be the output of a single run of the location privacy game. We say that the adversary *wins* if $d = b$, and we write it as $\mathsf{Exp}^{\mathsf{LocPriv}}_{\mathsf{DB}}(\mathcal{A}, 1^\lambda) = 1$.

The adversary can be considered as a hypothesis test for the following hypotheses:

$\mathcal{H}_0$: the response sent from the prover $\mathcal{P}_{\mathsf{Chal}}$ to $\mathcal{V}$'s challenge is actually from the prover P_0.

and

$\mathcal{H}_1$: the response sent from the prover $\mathcal{P}_{\mathsf{Chal}}$ to $\mathcal{V}$'s challenge is actually from the prover P_1.

We define the advantage of the adversary in this game as:

$$\mathsf{Adv}^{\mathsf{LocPriv}}_{\mathsf{DB}}(\mathcal{A}) = \left| 2\,\mathrm{Prob}\left[\mathsf{Exp}^{\mathsf{LocPriv}}_{\mathsf{DB}}(\mathcal{A}, 1^\lambda) = 1 \right] - 1 \right|.$$

Intuition: Location Privacy. There are a few essential points we highlight in our location privacy game. Firstly, note that we assume that the adversary knows the position of the verifier. This is not an unreasonable assumption, since in practice verifiers are often static and publicly known. In logistics, these verifiers are placed, for instance, at the entrance or exits of warehouses or shops. In public transport, the verifiers are placed at standard locations in means of public transport, and also within stations and stops. In personal identification or authentication, the verifiers are usually associated with doors. Thus, the assumption that the adversary knows the verifier's location and can choose its own location with respect to it.
A second assumption we make is that the adversary is able to eavesdrop on the communication between the prover and the verifier by eavesdropping on the timed and timeless channels. This assumption is standard in authentication scenarios, and we also use it in our basic, round-based framework, as presented in Chapter 2. We have also already discussed the assumption that the prover and verifiers both use clocks.
Finally, we comment on the approach to let the adversary generate copies of a prover at any location it wishes, before it distinguishes between two provers. This is a standard approach in distinguishing games, and it models the fact that there exists a particular combination of locations between which the adversary can distinguish: thus it is able to learn more about

the prover's location than that it is in the verifier's proximity. This is a stronger model than one in which the adversary can generate provers, but not dictate their location. We argue that in cryptography it is better to err on the safe side, and consider the worst case scenario.

6.3 Why Location Privacy does not Work

In this section we first argue that location privacy cannot be achieved with respect to an omniscient adversary. Then, we show that location privacy can only be achieved with respect to limited adversaries if the honest parties running the protocol introduce a delay in their transmissions; we furthermore give a lower bound on this delay.

6.3.1 Omniscient Adversary

It is trivial to see that no location privacy can be attained with respect to an omniscient adversary. Indeed, consider an omniscient adversary placed arbitrarily with respect to the verifier. Let this adversary $\mathcal{A}$ create two provers $\mathcal{P}_0$ and $\mathcal{P}_1$ such that the distance between this adversary and the provers is different,i.e., $t_{\mathcal{P}_0\mathcal{A}} \neq t_{\mathcal{P}_1\mathcal{A}}$. The adversary forwards $\mathcal{P}_0, \mathcal{P}_1$ to the challenger, receiving the handle $\mathcal{P}_{\mathsf{Chal}}$, which is either $\mathcal{P}_0$ or $\mathcal{P}_1$. Now, the adversary eavesdrops on a session between $\mathcal{P}_{\mathsf{Chal}}$ and $\mathcal{V}$, thus learning the sending time of the messages and the time it receives them. It thus calculates the time distance between itself and the two parties communicating and, since the distances are all different, it can identify the parties w.p. 1.
A single, but *moving* adversary (i.e., an adversary than can change its position during the attack) could also infer some information about the location of the prover by standing between $\mathcal{P}_0$ and $\mathcal{P}_1$ and moving toward $\mathcal{P}_0$. If bits arrive faster, they must be sent by $\mathcal{P}_0$ instead of $\mathcal{P}_1$.

6.3.2 Limited Adversary

By eavesdropping on the duplex timed channel between the challenge prover and the verifier, the adversary will receive $\mathsf{tr}^i_{\mathcal{A}}$, the time stamp when $\mathcal{A}$ receives the first bit of message m_i. The adversary $\mathcal{A}$ also observes:

- $t_{\mathcal{V}} = \mathsf{tr}^1_{\mathcal{A}}$: the time $\mathcal{A}$ receives the first message bit from $\mathcal{V}$.
- $t_{\mathcal{P}} = \mathsf{tr}^2_{\mathcal{A}}$: the time $\mathcal{A}$ receives the first message bit from $\mathcal{P}$.

In what follows we show that the very first bit sent through the timed channel leaks location-related information. To be able to prove that, we make the following (reasonable) assumptions as for how the sending time of this first bit is decided during the protocol. Note that similar observations hold for the final bit sent. For simplicity, we only treat the first one.

Assumption 6.2 *We assume that the distance bounding phase of a distance-bounding protocol may have one of the following constructions:*

- **Case 1:** *The verifier $\mathcal{V}$ starts the distance bounding phase after a reference time t_0 and a random delay, possibly equal to 0, which we denote $delay_{\mathcal{V}}$, while the prover $\mathcal{P}_b$ where $b \in \{0,1\}$ starts after receiving the first message from the verifier $\mathcal{V}$ and a random delay $delay_{\mathcal{P}_b}$.*
- **Case 2:** *The prover $\mathcal{P}_b$ starts the distance bounding phase after a reference time t_0 and a random delay $delay_{\mathcal{P}_b}$, while the verifier $\mathcal{V}$ starts after receiving the first message from the prover $\mathcal{P}_b$ and a random delay $delay_{\mathcal{V}}$.*
- **Case 3:** *The prover $\mathcal{P}_b$ and the verifier $\mathcal{V}$ start sending messages independently. More precisely, the prover $\mathcal{P}_b$ starts sending messages after a reference time $T_{\mathcal{P}_b}$ and a random delay $delay_{\mathcal{P}_b}$, while the verifier $\mathcal{V}$ starts sending messages after a reference time $T_{\mathcal{V}}$ and a random delay $delay_{\mathcal{V}}$.*

Assumption 6.3 *We also assume that $\mathcal{A}$ knows the times $T_{\mathcal{P}_b}$ (where $b \in \{0,1\}$) and $T_{\mathcal{V}}$; the latter value is defined only for Case 3 of* Assumption 6.2.

It is easy to see that in our model a limited adversary, $\mathcal{A}$ knows and can even choose the locations of $\mathcal{P}_0$, $\mathcal{P}_1$ with respect to itself and the verifier $\mathcal{V}$, i.e. the values $t_{\mathcal{A}\mathcal{P}_0}, t_{\mathcal{A}\mathcal{P}_1}, t_{\mathcal{V}\mathcal{P}_0}, t_{\mathcal{V}\mathcal{P}_1}$. Also, $\mathcal{A}$ knows the distance $t_{\mathcal{A}\mathcal{V}}$ to $\mathcal{V}$. We will show how an adversary intercepting the values above can distinguish between the two hypotheses with non-negligible probability.

Lemma 6.4 *Under Assumptions 6.2 and 6.3 we assume that there exists ϵ and a bound B such that:*

$$\mathbb{P}[delay \leq B] = 1 - \epsilon,$$

where delay might represent the delays of the provers $delay_{\mathcal{P}_0}$, $delay_{\mathcal{P}_1}$, or the delay ($delay_{\mathcal{V}}$) of the verifier as defined in Assumption 6.2 and $t_{\max}$ is the maximum allowed transmission time between a legitimate prover $\mathcal{P}$ and verifier $\mathcal{V}$. Then, there exists a passive adversary $\mathcal{A}$ against location indistinguishability which achieves a distinguishing advantage:

$$\mathsf{Adv}_{\mathcal{A}} \geq \left\lceil \frac{t_{\max}}{4B} \right\rceil (1 - 2\epsilon)$$

Assuming that the protocol is complete and polynomially bounded, there is a negligible ϵ such that B exists and is polynomially bounded. So, the advantage is significant.

Proof. Based on *Assumption 6.2* we have three cases.
Case 1: *The verifier $\mathcal{V}$ starts the* distance bounding phase *after a reference time t_0 and a random delay (denoted as $delay_{\mathcal{V}}$), whereas the prover $\mathcal{P}_b$ starts after receiving the first message from the verifier $\mathcal{V}$ and a random delay (denoted as $delay_{\mathcal{P}_b}$).*
We consider that the following events take place:

1. After some time reference t_0 and a $delay_{\mathcal{V}}$ the verifier $\mathcal{V}$ sends a message c to the prover $\mathcal{P}_b$ where $b \in \{0, 1\}$. The first bit of this message will arrive at the adversary $\mathcal{A}$ at time $t_{\mathcal{V}}$ such that:

$$t_{\mathcal{V}} = t_0 + delay_{\mathcal{V}} + t_{\mathcal{V}\mathcal{A}} \tag{1}$$

   where $t_{\mathcal{V}\mathcal{A}}$ denotes the time of flight for 1 bit from the verifier $\mathcal{V}$ to the adversary $\mathcal{A}$.

2. The prover $\mathcal{P}_b$ with $b \in \{0, 1\}$ responds to the verifier $\mathcal{V}$ with a message r, after some delay ($delay_{\mathcal{P}_b}$). The first bit of r arrives at $\mathcal{A}$ at time $t_{\mathcal{P}_b}$ such that:

$$t_{\mathcal{P}_b} = t_0 + delay_{\mathcal{V}} + t_{\mathcal{V}\mathcal{P}_b} + delay_{\mathcal{P}_b} + t_{\mathcal{P}_b\mathcal{A}} \tag{2}$$

   where $t_{\mathcal{V}\mathcal{P}_b}$ denotes the time-of-flight for one bit from $\mathcal{V}$ to $\mathcal{P}_b$, and $t_{\mathcal{P}_b\mathcal{A}}$ denotes the time-of-flight for one bit from the $\mathcal{P}_b$ to $\mathcal{A}$.

From equations (1) and (2) it is easy to see that:

$$t_{\mathcal{P}_b} - t_{\mathcal{V}} = t_{\mathcal{V}\mathcal{P}_b} - t_{\mathcal{V}\mathcal{A}} + delay_{\mathcal{P}_b} + t_{\mathcal{P}_b\mathcal{A}}$$

We let d_b be the probability density function (pdf) of $delay_{\mathcal{P}_b}$, i.e. we consider the delay to be a random variable distributed according to d_b. If hypothesis $\mathcal{H}_0$ holds, then $t_{\mathcal{P}} = t_{\mathcal{P}_0}$, while if hypothesis $\mathcal{H}_1$ holds, then $t_{\mathcal{P}} = t_{\mathcal{P}_1}$. Since $t_{\mathcal{P}}$ and $t_{\mathcal{V}}$ depend on random delays, they can be perceived as random variables. Let:

$$T = t_{\mathcal{P}} - t_{\mathcal{V}} - t_{\mathcal{V}\mathcal{P}_0} + t_{\mathcal{V}\mathcal{A}} - t_{\mathcal{P}_0\mathcal{A}} \quad \text{and } \Delta = t_{\mathcal{V}\mathcal{P}_1} + t_{\mathcal{P}_1\mathcal{A}} - t_{\mathcal{V}\mathcal{P}_0} - t_{\mathcal{P}_0\mathcal{A}}$$

Note that whereas the value Δ is fixed and even chosen by the adversary, T is a random variable, depending on the delays. Indeed, if hypothesis $\mathcal{H}_0$ holds then $T = delay_{\mathcal{P}_0}$ has pdf d_0, while if hypothesis $\mathcal{H}_1$ holds, then $T = delay_{\mathcal{P}_1} + \Delta$ and we write $\mathrm{Prob}[T = t] = d_1(t - \Delta)$, i.e. T has a distribution equivalent to d_1, shifted by a fixed value Δ.
In the following, we often condition success probabilities on hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ and use the notation $\mathbb{P}_{\mathcal{H}_b}[\text{event}]$ for $\mathbb{P}[\text{event} \mid \mathcal{H}_b \text{ holds}]$, i.e. the probability that *event* holds, conditioned on the fact that $\mathcal{H}_b$ holds.
We consider that $\mathcal{A}$ is implementing a best distinguisher based on the likelihood that $\mathbb{P}_{\mathcal{H}_0}[T = t] > \mathbb{P}_{\mathcal{H}_1}[T = t]$ for observed value t. If this holds, then $\mathcal{A}$ outputs 0, else it outputs 1. So $\mathcal{A}$ outputs 0 iff. the observed value of $T = t_{\mathcal{P}} - t_{\mathcal{V}} - t_{\mathcal{V}\mathcal{P}_0} + t_{\mathcal{V}\mathcal{A}} - t_{\mathcal{P}_0\mathcal{A}}$ is $T = t$ such that:

$$\mathbb{P}[t = delay_{\mathcal{P}_0}] > \mathbb{P}[t = delay_{\mathcal{P}_1} + \Delta]$$

Then, it holds:

$$\mathsf{Adv} = \mathbb{P}_{\mathcal{H}_0}[\mathcal{A} \to 0] - \mathbb{P}_{\mathcal{H}_1}[\mathcal{A} \to 0] = \frac{1}{2} \int_{-\infty}^{+\infty} |d_0(t) - d_1(t - \Delta)| dt, \tag{3}$$

where d_0 and d_1 make $[0, B]$ have density at least $1 - \epsilon$. When $t_{\mathcal{P}_0\mathcal{V}} = t_{\mathcal{P}_1\mathcal{V}} = t_{\max}$, $\mathcal{P}_0$, $\mathcal{V}$ and $\mathcal{P}_1$ are aligned in this order and the adversary $\mathcal{A}$ overlaps with the location of $\mathcal{P}_0$, then $\Delta = 2t_{\max}$.

Case 2: *The prover $\mathcal{P}_b$ starts the* distance bounding phase *after a reference time t_0 and a random delay (denoted as $delay_{\mathcal{P}_b}$). While the verifier $\mathcal{V}$ starts after receiving the first message from the prover $\mathcal{P}_b$ and a random delay (denoted as $delay_{\mathcal{V}}$).*

Now, we have:

$$\begin{aligned} t_{\mathcal{P}_b} &= t_0 + delay_{\mathcal{P}_b} + t_{\mathcal{P}_b\mathcal{A}} \\ t_{\mathcal{V}} &= t_0 + delay_{\mathcal{P}_b} + t_{\mathcal{P}_b\mathcal{V}} + delay_{\mathcal{V}} + t_{\mathcal{V}\mathcal{A}} \\ t_{\mathcal{V}} - t_{\mathcal{P}_b} &= t_{\mathcal{P}_b\mathcal{V}} + delay_{\mathcal{V}} + t_{\mathcal{V}\mathcal{A}} - t_{\mathcal{P}_b\mathcal{A}} \end{aligned}$$

We let:

$$T = t_{\mathcal{V}} - t_{\mathcal{P}} - t_{\mathcal{P}_0\mathcal{V}} - t_{\mathcal{V}\mathcal{A}} + t_{\mathcal{P}_0\mathcal{A}} \quad \text{and } \Delta = t_{\mathcal{P}_1\mathcal{V}} - t_{\mathcal{P}_1\mathcal{A}} - t_{\mathcal{P}_0\mathcal{V}} + t_{\mathcal{P}_0\mathcal{A}}$$

Similarly, if the adversary $\mathcal{A}$ is implementing a distinguisher for the two provers $\mathcal{P}_0$ and $\mathcal{P}_1$ then its advantage is given by:

$$\mathsf{Adv} = \mathbb{P}_{\mathcal{H}_0}[\mathcal{A} \to 0] - \mathbb{P}_{\mathcal{H}_1}[\mathcal{A} \to 0] = \frac{1}{2}\int_{-\infty}^{+\infty} |d(t) - d(t - \Delta)|dt \tag{4}$$

where d denotes the pdf of the random variable $delay_{\mathcal{V}}$, such that $[0, B]$ has density at least $1 - \epsilon$. When $t_{\mathcal{P}_0\mathcal{V}} = t_{\mathcal{P}_1\mathcal{V}} = t_{\max}$, $\mathcal{P}_0$, $\mathcal{V}$ and $\mathcal{P}_1$ are aligned and the location of the adversary $\mathcal{A}$ overlaps with the location of the prover $\mathcal{P}_1$, then $\Delta = 2t_{\max}$. Thus, from equations (3) and (4) we derive that in both cases it holds:

$$\mathsf{Adv} = \frac{1}{2}\int_{-\infty}^{+\infty} |q_0(t) - q_1(t - \Delta)|dt$$

for some functions q_0 and q_1 that make $[0, B]$ have density at least $1 - \epsilon$. We further have a case where $\Delta = 2t_{\max}$.
Let:

$$x_{b,i} = \int_{(i-1)|\Delta|}^{i|\Delta|} q_b(t)dt \quad \text{and } n = \left\lceil \frac{B}{|\Delta|} \right\rceil$$

We have $x_{b,0} = 0$, $x_{b,n+1} = 0$, $x_{b,i} \geq 0$ and $x_{b,1} + \cdots + x_{b,n} \geq 1 - \epsilon$. Given $I \subseteq \{0, \ldots, n\}$ we let $T_I = \bigcup_{i \in I} \big[(i-1)|\Delta|, i|\Delta|\big]$. For $\Delta > 0$, we have:

$$\mathsf{Adv}_{T_I,\Delta} = \sum_{i \in I}(x_{0,i} - x_{1,i-1}) \text{ and } \mathsf{Adv}_{T_I,-\Delta} = \sum_{i \in I}(x_{0,i} - x_{1,i+1})$$

Let:

$$\mathsf{Adv}_{\Delta} = \max_I \mathsf{Adv}_{T_I,\Delta} = \frac{1}{2}\sum_{i=0}^{n} |x_{0,i} - x_{1,i-1}|$$

$$\mathsf{Adv}_{-\Delta} = \max_I \mathsf{Adv}_{T_I,-\Delta} = \frac{1}{2}\sum_{i=0}^{n} |x_{0,i} - x_{1,i+1}|$$

We have:

$$\mathsf{Adv}_{\Delta} + \mathsf{Adv}_{-\Delta} = \frac{1}{2}\sum_{i=0}^{n}(|x_{0,i} - x_{1,i-1}| + |x_{0,i} - x_{1,i+1}|) \geq \frac{1}{2}\sum_{i=0}^{n} |x_{1,i+1} - x_{1,i-1}|$$

Since $x_{1,i} \geq 0$ and $x_{1,1} + \cdots + x_{1,n} \geq 1 - \epsilon$, there exists j such that: $x_{1,j} \geq \frac{1-\epsilon}{n}$. Thus:

$$\mathsf{Adv}_{\Delta} + \mathsf{Adv}_{-\Delta} \geq \frac{1}{2}(|x_{1,j} - x_{1,j-2}| + |x_{1,j-2} - x_{1,j-4}| + \ldots) \geq \frac{x_{1,j}}{2} \geq \frac{1-\epsilon}{2n}$$

Thus,

$$\max(\mathsf{Adv}_{\Delta}, \mathsf{Adv}_{-\Delta}) \geq \frac{1-\epsilon}{4n}$$

So, there exists Δ such that:

$$\mathsf{Adv}_{\Delta} \geq \left\lceil \frac{|\Delta|}{4B} \right\rceil (1-\epsilon)$$

For $\Delta = 2t_{\max}$ there exists an adversary $\mathcal{A}$ such that:

$$\mathsf{Adv}_{\mathcal{A}} \geq \left\lceil \frac{t_{\max}}{2B} \right\rceil (1-\epsilon)$$

Case 3: *The prover $\mathcal{P}_b$ and the verifier $\mathcal{V}$ send messages independently. More precisely, the prover $\mathcal{P}_b$ starts sending messages after a reference time $T_{\mathcal{P}_b}$ and a random delay ($delay_{\mathcal{P}_b}$) while the verifier $\mathcal{V}$ starts sending messages after a reference time $T_{\mathcal{V}}$ and a random delay ($delay_{\mathcal{V}}$). We assume that for this case the adversary $\mathcal{A}$ knows the values $T_{\mathcal{P}_b} - T_{\mathcal{V}}$.*

We now have:

$$\begin{aligned} t_{\mathcal{V}} &= T_{\mathcal{V}} + delay_{\mathcal{V}} + t_{\mathcal{V}\mathcal{A}} \\ t_{\mathcal{P}_b} &= T_{\mathcal{P}_b} + delay_{\mathcal{P}_b} + t_{\mathcal{P}_b\mathcal{A}} \\ t_{\mathcal{P}_b} - t_{\mathcal{V}} &= delay_{\mathcal{P}_b} - delay_{\mathcal{V}} + T_{\mathcal{P}_b} + t_{\mathcal{P}_b\mathcal{A}} - T_{\mathcal{V}} - t_{\mathcal{V}\mathcal{A}} \end{aligned}$$

We let:

$$T = t_{\mathcal{P}} - t_{\mathcal{V}} - T_{\mathcal{P}_1} - t_{\mathcal{P}_1\mathcal{A}} + T_{\mathcal{V}} + t_{\mathcal{V}\mathcal{A}} \text{ and } \Delta = T_{\mathcal{P}_1} + t_{\mathcal{P}_1\mathcal{A}} - T_{\mathcal{P}_0} - t_{\mathcal{P}_0\mathcal{A}} \tag{5}$$

We consider that the adversary $\mathcal{A}$ is implementing a best distinguisher based on the likelihood if $\mathbb{P}_{\mathcal{H}_0}[t_{\mathcal{P}} - t_{\mathcal{V}}] > \mathbb{P}_{\mathcal{H}_1}[t_{\mathcal{P}} - t_{\mathcal{V}}]$ then $\mathcal{A}$ outputs 0 otherwise it outputs 1. So, $\mathcal{A}$ outputs 0 iff $t_{\mathcal{P}} - t_{\mathcal{V}} - T_{\mathcal{P}_1} - t_{\mathcal{P}_1\mathcal{A}} + T_{\mathcal{V}} + t_{\mathcal{V}\mathcal{A}} = T = t$ such that:

$$\mathcal{P}[t = delay_{\mathcal{P}_0} - delay_{\mathcal{V}}] > \mathcal{P}[t = delay_{\mathcal{P}_1} - delay_{\mathcal{V}} + \Delta]$$

Then, it holds:

$$\mathsf{Adv} = \mathbb{P}_{\mathcal{H}_0}[\mathcal{A} \to 0] - \mathbb{P}_{\mathcal{H}_1}[\mathcal{A} \to 0] = \frac{1}{2} \int_{-\infty}^{+\infty} |q_0(t) - q_1(t-\Delta)| dt \tag{6}$$

where q_b for $b \in \{0,1\}$ denotes the pdf of the random variable $delay_{\mathcal{P}_b} - delay_{\mathcal{V}}$ and the support of q_0 and q_1 make $[-B, B]$ have density at least $1 - 2\epsilon$. When $t_{\mathcal{P}_0\mathcal{V}} = t_{\mathcal{P}_1\mathcal{V}} = t_{\max}$, $\mathcal{P}_0$, $\mathcal{V}$ and $\mathcal{P}_1$ are aligned in this order and if $T_{\mathcal{P}_1} \geq T_{\mathcal{P}_0}$ the location of the adversary $\mathcal{A}$ overlaps with the location of $\mathcal{P}_0$ while if $T_{\mathcal{P}_1} < T_{\mathcal{P}_0}$ the location of the adversary $\mathcal{A}$ overlaps with the location of the prover $\mathcal{P}_1$. Thus, in both of these cases it holds that $|\Delta| \geq 2t_{\max}$. Let:

$$x_{b,i} = \int_{(i-1)|\Delta|}^{i|\Delta|} q_b(t) dt \quad \text{and } n = \left\lceil \frac{B}{|\Delta|} \right\rceil$$

We have $x_{b,0} = 0$, $x_{b,n+1} = 0$, $x_{b,i} \geq 0$, $x_{b,-n+1} + ... + x_{b,n} \geq 1 - 2\epsilon$ and:

$$\mathsf{Adv}_{\Delta} + \mathsf{Adv}_{-\Delta} = \frac{1}{2} \sum_{i=-n}^{n} (|x_{0,i} - x_{1,i-1}| + |x_{0,i} - x_{1,i+1}|) \geq \frac{1}{2} \sum_{i=0}^{-n} |x_{1,i+1} - x_{1,i-1}|$$

Since $x_{1,i} \geq 0$ and $x_{1,-n+1} + \cdots + x_{1,n} \geq 1 - 2\epsilon$, there exists j such that: $x_{1,j} \geq \frac{1-2\epsilon}{2n}$. Thus:

$$\mathsf{Adv}_{\Delta} + \mathsf{Adv}_{-\Delta} \geq \frac{1}{2}(|x_{1,j} - x_{1,j-2}| + |x_{1,j-2} - x_{1,j-4}| + \dots) \geq \frac{x_{1,j}}{2} \geq \frac{1-2\epsilon}{4n}$$

Thus,

$$\max(\mathsf{Adv}_{\Delta}, \mathsf{Adv}_{-\Delta}) \geq \frac{1-2\epsilon}{8n}$$

So, there exists Δ such that:

$$\mathsf{Adv} \geq \left\lceil \frac{|\Delta|}{8B} \right\rceil \geq \frac{t_{\max}}{4B}(1-2\epsilon)$$

□

Lemma 6.5 *If Assumption 6.2 holds and d_b follows the uniform distribution in the range $[0, B]$ and denotes the pdf of the $delay_{\mathcal{P}_b}$ while the $delay_{\mathcal{V}}$ is always equal to 0, then the best distinguisher based on $t_{\mathcal{P}} - t_{\mathcal{V}}$ and the locations satisfies:*

$$\mathsf{Adv}_{\mathcal{A}} = \frac{2t_{\max}}{B}$$

Proof. Following the proof of the Lemma 6.4 the best distinguisher based on $t_{\mathcal{P}} - t_{\mathcal{V}}$ and the locations (of the provers and the verifier) follows equations (3), (4), or (6). So, it satisfies:

$$\mathsf{Adv} = \frac{1}{2} \int_{-\infty}^{+\infty} |d_0(t) - d_1(-\Delta + t)| \, dt$$

since $delay_{\mathcal{V}} = 0$. Since d_b follows the uniform distribution in the range $[0, B]$, it holds:

$$\mathsf{Adv}_{\mathcal{A}} = \frac{1}{2} \int_0^{\Delta} \frac{dt}{B} + \frac{1}{2} \int_B^{B+\Delta} \frac{dt}{B} = \frac{\Delta}{B}$$

and Δ is bounded by $2t_{\max}$ in all three cases.

□

Practical Consequences: Although the attack is polynomial, we can still live with it in practice thanks to the very high celerity of light, since the time it takes to cover 10m is 2^{-25} sec. Indeed, let:

$$h = \log_2 \frac{B}{2t_{\max}}$$

The best advantage is comparable to guessing h bits correctly. To have a privacy level of h bits (i.e., a best advantage of 2^{-h}), we shall thus have:

$$B \geq 2^{h+1} t_{\max}$$

For instance, when $t_{\max}$ is the time light takes to go though the distance of 10 m and $h = 20$ bits (i.e., an adversary cannot distinguish two provers, accept with one chance out of a million), we have $B \geq 0.07$ sec, which is still a reasonable delay.

6.4 Location Private Construction

In this section we apply our results from the previous section to achieve a location private distance-bounding protocol for limited adversaries. The proposed protocol is based on the RČ protocol [69]. We first review the original RČ protocol, noting that the construction is vulnerable to a mafia fraud attack, which we show in Section 6.4.1. Having improved the construction so as to make it mafia fraud secure, we also apply our results of the previous section to attain the first provably location private distance-bounding protocol in the literature.

6.4.1 The RČ Distance-Bounding Protocol

In what follows, we outline the distance-bounding protocol due to Rasmussen and Čapkun [69]. This protocol runs in two consecutive phases, at the end of which the verifier $\mathcal{V}$ upper-bounds the distance between itself and the prover $\mathcal{P}$. The two parties share two secret keys K_1 and K_2, using one for encryption and the other for signing. Initially, the prover and verifier run the so-called *initialisation phase*, where the hidden marker M is generated by the verifier and sent (as part of an encryption) to the prover. Subsequently, the two parties run the *distance-bounding* phase.

The RČ Protocol. The protocol uses a symmetric encryption scheme (KGen, Enc, Dec), and an unforgeable signature scheme (Setup, Sign, Verify).

- **Initialisation Phase:**
  - **Step 1:** The prover $\mathcal{P}$ generates a random nonce $N_{\mathcal{P}}$ of length n. It then computes the encryption $\mathsf{Enc}_K(\mathcal{P}, \mathcal{V}, N_{\mathcal{P}})$ where $\mathcal{P}$ and $\mathcal{V}$ denotes the identities of $\mathcal{P}$ and $\mathcal{V}$ correspondingly. The prover also calculates the MAC of N_P (i.e. $MAC(N_{\mathcal{P}})$) and sends the concatenation of the two values (i.e. $c_1 = Enc_K(\mathcal{P}, \mathcal{V}, N_{\mathcal{P}}) \| MAC(N_{\mathcal{P}})$) to the verifier $\mathcal{V}$.
  - **Step 2:** $\mathcal{V}$ receives the value c_1, generates a random variable M of length m, called *hidden marker*, and computes the encryption $\mathsf{Enc}_K(\mathcal{P}, \mathcal{V}, N_{\mathcal{P}})$ as well as the MAC of the concatenation of $N_{\mathcal{P}}$ and M (i.e. $MAC(N_{\mathcal{P}} \| M)$. Finally he sends the concatenation of the two results (i.e. $c_2 = \mathsf{Enc}_K(\mathcal{P}, \mathcal{V}, N_{\mathcal{P}}) \| MAC(N_{\mathcal{P}} \| M)$). Finally the verifier $\mathcal{V}$ generates a random nonce $N_{\mathcal{V}}$ of length n.
  - **Step 3:** $\mathcal{P}$ parses the response as $(\hat{c}_2 \| \hat{t}_2)$. It then decrypts the value $\hat{c}_2$ and checks that the obtained plaintext verifies for the signature $\hat{t}_2$. If so, the distance bounding phase is run as below; else, the protocol aborts.
- **Distance Bounding (DB) Phase:** In this phase $\mathcal{P}$ and $\mathcal{V}$ transmit simultaneously, in a constant bit stream. The verifier $\mathcal{V}$ transmits a stream $stream_{\mathcal{V}}$ as follows:

$$stream_{\mathcal{V}} := Rand_{\mathcal{V}_1} \| M \| N_{\mathcal{V}} \| Rand_{\mathcal{V}_2}.$$

The beginning time of this transmission is not clearly defined in the RČ protocol. However, it seems that this bit stream is transmitted simultaneously with a stream $stream_{\mathcal{P}}$ generated by the prover $\mathcal{P}$ such that:

$$\begin{aligned} stream_{\mathcal{P}} &:= Rand_{\mathcal{V}_1} \oplus Rand_{\mathcal{P}_1} \| \hat{M} \oplus Rand_{\mathcal{P}_2} \| \\ &\quad \hat{N}_{\mathcal{V}} \oplus N_{\mathcal{P}} \| Rand_{\mathcal{V}_2} \oplus Rand_{\mathcal{P}_3} \\ &:= Rand_{\mathcal{P}_4} \| \hat{N}_{\mathcal{V}} \oplus N_{\mathcal{P}} \| Rand_{\mathcal{P}_5}, \end{aligned}$$

Here, it holds that $Rand_{\mathcal{P}_4} := Rand_{\mathcal{V}_1} \oplus Rand_{\mathcal{P}_1}$ and $Rand_{\mathcal{P}_5} := Rand_{\mathcal{V}_2} \oplus Rand_{\mathcal{P}_3}$. We note that the prover $\mathcal{P}$ parses the received bits from the stream $stream_{\mathcal{V}}$ and sends its own transmission of $stream_P$ at the same time; however it is not clear how the two parties know when to begin their simultaneous continuous bit stream exchange. The distance-bounding properties of the protocol rely on the fact that $\mathcal{P}$'s response is asynchronous; this, however, is also not clearly stated in the RČ protocol description. We describe this process in five steps:

1. $\mathcal{V}$ generates and sends the random data $Rand_{\mathcal{V}_1}$ to $\mathcal{P}$. As $\mathcal{P}$ receives this data, it XORs the received bits with random data generated by itself and responds with the resulting stream $(Rand_{\mathcal{V}_1} \oplus Rand_{\mathcal{P}_1})$. Depending on when $\mathcal{P}$ really starts, some leading bits of $Rand_{\mathcal{V}1}$ may be ignored by $\mathcal{P}$ (if $\mathcal{P}$ starts later) or some extra leading 0's may be added by $\mathcal{P}$ (if $\mathcal{P}$ starts earlier).
2. At some randomly selected point, unspecified by [69], $\mathcal{V}$ starts transmitting the hidden marker M, which the $\mathcal{P}$ also XORs with random data $(\hat{M} \oplus Rand_{\mathcal{P}_2})$, sending this as part of its stream.
3. After M is fully transmitted, $\mathcal{V}$ starts sending $N_{\mathcal{V}}$ (i.e. the nonce it generated during the initialisation phase) to $\mathcal{P}$. The prover, who has also computed the hidden marker M, will expect $N_{\mathcal{V}}$ to be transmitted after the transmission of M. When M stops, the prover XORs the subsequent received bits (which we denote $\hat{N}_{\mathcal{V}}$) with its own random nonce $N_{\mathcal{P}}$, sending in its continuous stream the value $(\hat{N}_V \oplus N_{\mathcal{P}})$.
4. After finishing the transmission of $N_{\mathcal{V}}$, the verifier $\mathcal{V}$ restarts transmitting random data $Rand_{\mathcal{V}_2}$, to which the prover $\mathcal{P}$ responds by XORing this data with random values of its own as follows: $Rand_{\mathcal{V}_2} \oplus Rand_{\mathcal{P}_3}$. Both parties continue sending random data for a random interval, then stop transmitting. Again, depending on whether $\mathcal{P}$ halts before or after $\mathcal{V}$, some tailing bits are ignored or some extra are added by $\mathcal{P}$.
5. At the end of this phase, the verifier $\mathcal{V}$ counts the number of bits it received between sending the first bit of $N_{\mathcal{V}}$ and receiving the first bit of the value $N_{\mathcal{V}} \oplus \hat{N}_{\mathcal{P}}$. This delay can be translated into an upper bound on the distance $\Delta t(\mathcal{P}, \mathcal{V})$ by using the bit rate and the process delay. The entire protocol is depicted in detail in Figure 29.

Notes on the protocol: Note that the XORing process is wholly unnecessary in this protocol: in fact, the prover could simply respond by transmitting random data and, after the hidden marker M, it could simply reply with the value $N_{\mathcal{P}}$. If the messages must necessarily be both and MACed, we suggest using state-of-the-art symmetric authenticated encryption.

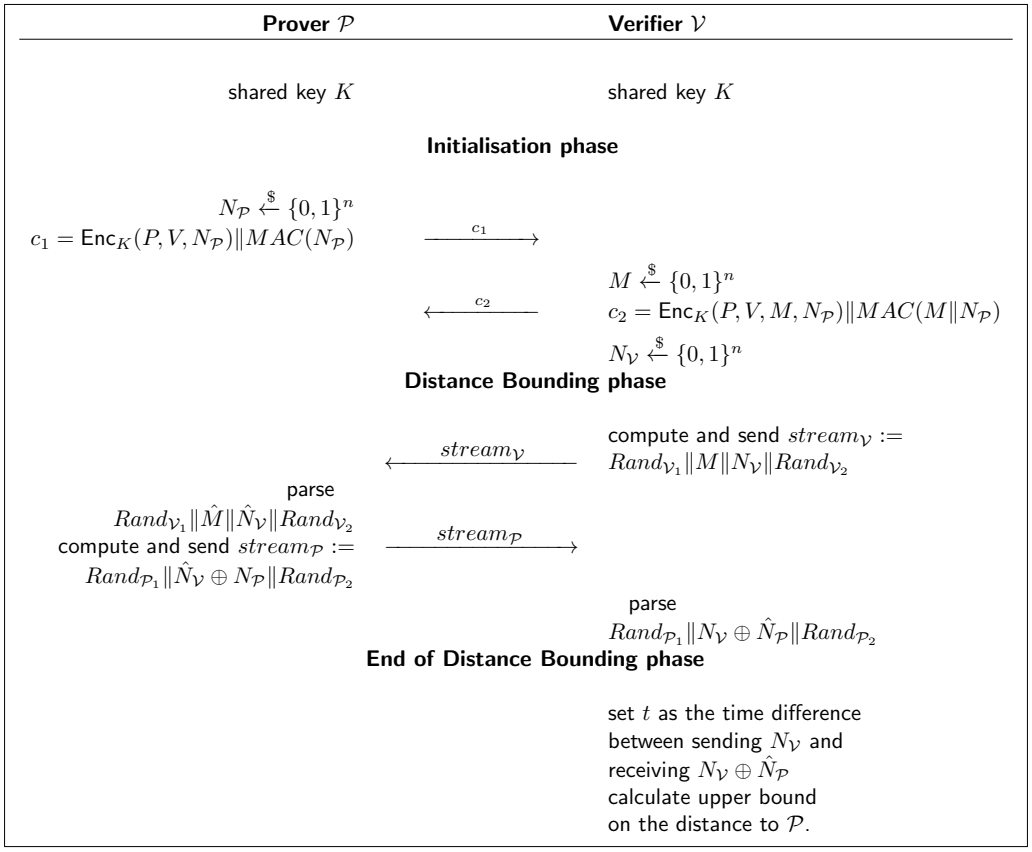

Figure 29: The RČ protocol. Here $\xleftarrow{\$}$ denotes sampling uniformly at random. Note that during the distance-bounding phase, the $\rightarrow$ transmission in constant stream, whereas during the initialisation phase, this is not the case

A mafia-fraud attack. In the original description of the RČ protocol in [69], Rasmussen and Čapkun state that their scheme is not to be used in authentication. Instead the protocol is meant to ensure "distance-bounding" and location privacy. Our results show that this protocol is in fact *not* location private, unless a delay as prescribed by our results is included in the protocol description.

Furthermore we note that the term distance-bounding in the literature is usually meant as an extension of authentication, which furthermore ensures that adversaries cannot perform MITM attacks. By contrast, distance fraud resistance is in our view a much weaker requirement, and it is unreasonable to use expensive primitives such as encryption and MAC schemes in order to achieve it. Indeed, distance fraud resistance can be achieved by simply echoing bits between the prover and the verifier.

If we consider distance-bounding to be defined as we show in Chapter 2, we assume, minimally, that the construction is resistant to distance and mafia fraud attacks. We show that this is not the case, since we can describe a mafia fraud attack that gives an attacker a non-negligible success probability in its impersonation attempt. We show this attack pictographically in Figure 30, and then describe it in more detail.

This attack can be divided in three stages: the Mafia fraud mode stage, the impersonation stage, and the repetition stage. More precisely:

Stage 1: Mafia fraud mode

In this stage, the adversary $\mathcal{A}$ acts as a man-in-the-middle (MITM) when the protocol is run between a legitimate prover $\mathcal{P}$ and a legitimate verifier $\mathcal{V}$.

More precisely, the prover $\mathcal{P}$ sends some value c_1 (i.e. $\mathsf{Enc}_K(\mathcal{P}, \mathcal{V}, N_{\mathcal{P}}) \| MAC(N_{\mathcal{P}})$) to the verifier $\mathcal{V}$. The adversary acting as a MITM relays c_1 to the verifier $\mathcal{V}$. The verifier $\mathcal{V}$ responds with a value c_2 (i.e. $\mathsf{Enc}_K(\mathcal{P}, \mathcal{V}, M, N_{\mathcal{V}}, N_{\mathcal{P}}) \| MAC(M \| N_{\mathcal{V}} \| N_{\mathcal{P}})$)

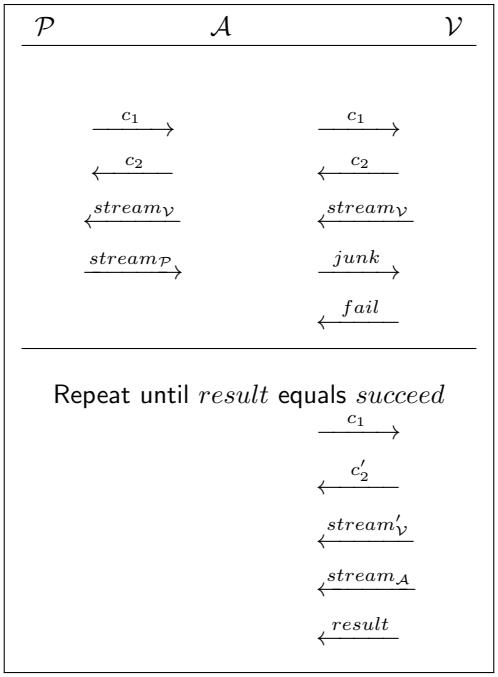

Figure 30: Attack scenario

where M, $N_{\mathcal{V}}$, $N_{\mathcal{P}}$ are used for the hidden marker and the random nonces generated by the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ receptively and the adversary relays these messages.
During the distance bounding phase, $\mathcal{A}$ relays $stream_{\mathcal{V}}$ to $\mathcal{P}$ and replies some junk stream to $\mathcal{V}$. The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ send the streams $stream_{\mathcal{P}}$ and resp. $stream_{\mathcal{V}}$ such that:

$$stream_{\mathcal{V}} = Rand_1 \| M \| N_{\mathcal{V}} \| Rand_2, \quad \text{and}$$
$$stream_{\mathcal{P}} = Rand_3 \| N_{\mathcal{V}} \oplus N_{\mathcal{P}} \| Rand_4.$$

where $Rand_i$, for $i \in \{1, 2, 3, 4\}$, denotes random data sent either by the prover $\mathcal{P}$ or the verifier $\mathcal{V}$. After this, it holds that:

$$(stream_{\mathcal{V}} \| \mathbf{0}) \oplus \mathsf{Shift}_p(stream_{\mathcal{P}} \| \mathbf{0}) = Rand_5 \| N_{\mathcal{P}} \| Rand_6 \| \mathbf{0} \tag{7}$$

where $\mathbf{0}$ denotes a bit stream of infinite length with only 0 bits, and p denotes a necessary offset which depends on when $\mathcal{P}$ and $\mathcal{V}$ start sending their transmission. Here, $\mathsf{Shift}_k(s)$ denotes a function that performs a shift of a stream s for k bits; k can be positive or negative and thus, the shift is performed right or left correspondingly. More precisely, the offset p depends on the time $t_{\mathcal{P}}$ at which the prover $\mathcal{P}$ starts its transmission, the time $t_{\mathcal{A}}$ at which the adversary $\mathcal{A}$ starts its transmission and the time $t_{\mathcal{V}\mathcal{P}}$ depending on the time distance $\Delta\mathsf{t}(\mathcal{P}, \mathcal{A})$ that is required for a message (consisting of possibly many bits) to be transmitted from the adversary $\mathcal{A}$ to the prover $\mathcal{P}$. Thus, the offset p is given by the following equation:

$$p = (t_{\mathcal{A}} - t_{\mathcal{A}\mathcal{P}} - t_{\mathcal{P}}) * f$$

where f denotes the number of bits sent per second during the distance bounding phase between the prover $\mathcal{P}$ and the verifier $\mathcal{V}$. We assume in this paper that f is the same for $\mathcal{P}$ and $\mathcal{V}$.
If we assume that the adversary $\mathcal{A}$ can physically observe the locations of $\mathcal{P}$ and $\mathcal{A}$, this means it can also deduce the times $t_{\mathcal{P}}$ and $t_{\mathcal{A}\mathcal{P}}$. Thus, it can calculate the value of p. If we assume an adversary $\mathcal{A}$ that only knows the location of the verifier, this adversary may just make a guess for p. Note that p is bounded by the length of $stream_{\mathcal{P}}$ and $stream_{\mathcal{V}}$, so it must be small in order for the protocol to be efficient.
The adversary $\mathcal{A}$ also makes a guess for the position L of $N_{\mathcal{P}}$ in the stream $stream_V \oplus \mathsf{Shift}_p(stream_{\mathcal{P}})$ (i.e. equation (7)) and deduces a value $N'_{\mathcal{P}}$ based on this guess. If the position L was guessed correctly, then the adversary can deduce $N_{\mathcal{P}}$ exactly.

$\mathcal{A}$'s state: At this point, the adversary has stored the values c_1, t_1, and the guesses for p, L and $N_{\mathcal{P}}$, which we denote p, L, and $N'_{\mathcal{P}}$. Note that stronger adversaries, who are aware of the prover's position, know the offset p and thus do not need to guess it.

Stage 2: Impersonation

This second stage is depicted in Figure 30 and labeled as "repeat". Here, the adversary $\mathcal{A}$ starts a new session with the verifier $\mathcal{V}$ and sends as its first message the eavesdropped values c_1. Note that this allows the adversary to replay the same $N_{\mathcal{P}}$, regardless of how secure the encryption and the signature schemes are.
The verifier $\mathcal{V}$ will generate its own (fresh) nonce and hidden marker. Thus, the values used in this session are M', $N_{\mathcal{P}}$, and $N'_{\mathcal{V}}$. In the distance bounding phase of this session, the verifier $\mathcal{V}$ sends a new stream of bits $stream'_V$ such that:

$$stream'_V = Rand'_1 \| M' \| N'_{\mathcal{V}} \| Rand'_2$$

In turn, the adversary $\mathcal{A}$ responds with its own bitstream $stream_{\mathcal{A}}$ computed as follows:

$$stream_{\mathcal{A}} = \mathsf{Shift}_q(stream'_V) \oplus (N'_{\mathcal{P}} \| N'_{\mathcal{P}} \| \ldots \| N'_{\mathcal{P}})$$

where $N'_{\mathcal{P}}$ denotes the value that the adversary $\mathcal{A}$ has guessed for $N_{\mathcal{P}}$ in Stage 1 and q is a required alignment (compensating for the distance between the verifier and the adversary). The alignment q must be chosen such that $\mathcal{A}$ is sure that the value $N'_{\mathcal{V}} \oplus N'_{\mathcal{P}}$ is included in $stream_{\mathcal{A}}$. In other words, the length (L') of the $Rand'_1$, i.e. the number of random bits transmitted in the impersonation session before the hidden marker M' is sent, is a multiple of the length of $N_{\mathcal{P}}$.
Thus, the verifier $\mathcal{V}$ should be able to calculate the value of the offset q such that it satisfies the following condition:

$$q = (|M'| + L') \bmod |N_{\mathcal{P}}|$$

Insight: the Offset q: The offset q need not be equal to the length (in bits) of $Rand'_1||M'$. Instead, we write $|(Rand'_1||M')| = k \cdot n + q$, where n is the length of the prover and verifier nonces. Since q is a remainder of the division by n, it follows that it can take values between 0 and $n-1$, and can be guessed with probability $\frac{1}{n}$. Consequently, if the offset q is guessed accurately, the verifier will receive the response $N'_{\mathcal{P}} \oplus N'_{\mathcal{V}}$ upon transmitting $N'_{\mathcal{V}}$. Thus, if the adversary's guess of $N_{\mathcal{P}}$ is accurate, the attack succeeds.

Attack Scenarios. The attack can be launched in different scenarios depending on the location of the legitimate prover $\mathcal{P}$. One scenario might include an adversary $\mathcal{A}$ whose goal is to impersonate (stage 2) a legitimate prover $\mathcal{P}$ after having eavesdropped (stage 1) a session between $\mathcal{P}$ and $\mathcal{V}$. In such a scenario the prover $\mathcal{P}$ is in the range of the verifier $\mathcal{V}$ (depicted in Figure 31, left figure).
Another scenario is our mafia fraud where the legitimate prover $\mathcal{P}$ might be located very far from the verifier $\mathcal{V}$ while the adversary is in the communication range of $\mathcal{V}$ (depicted in Figure 31, right figure). In this case the adversary $\mathcal{A}$ relays messages between $\mathcal{P}$ and $\mathcal{V}$ (i.e. corresponding to stage 1 of the described attack). Obviously, by simply relaying messages the distance bounding protocol will fail since the legitimate prover $\mathcal{P}$ is located quite far from the verifier $\mathcal{V}$. Nevertheless, the adversary $\mathcal{A}$ will be able to restart the protocol (i.e. stage 2 of the described attack) and use the information it received from Stage 1 to get authenticated.

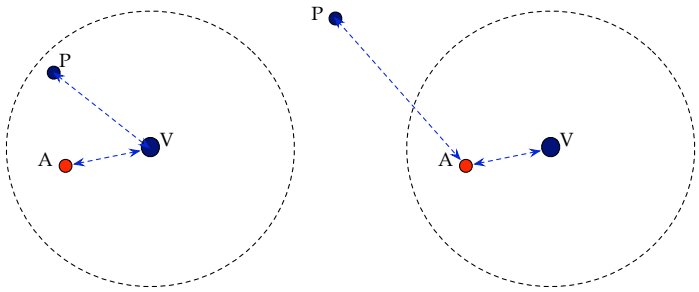

Figure 31: Attack Scenarios 1 (left) and 2 (right)

Success of the Attack: As described above, the success of the attack depends on calculating/guessing correctly the offsets p and q and the length L.

If we denote by P_s the success probability of the attack and by P_p, P_q and P_L the probability to successfully guess the offsets p and q and the length L correspondingly then it holds that:

$$P_s = P_p * P_q * P_L.$$

The value P_L depends on the min-entropy of the distribution of L in the protocol (a value unspecified in [69]). By guessing that $L = \arg\max_\ell \mathbb{P}[L = \ell]$, we have $P_L = 2^{-\mathcal{E}_{min}(L)}$, where $\mathcal{E}_{min}(L)$ is the min-entropy of L.
The distribution that minimises the $\mathcal{E}_{min}(L)$ over a given set of values is the uniform distribution. The adversary knows that the verifier *must* send the nonce $N_\mathcal{V}$, preceded by the hidden marker. Thus, its guess of L ranges over the values $\{0, 1, \ldots, |stream_V| - |N_\mathcal{V}|\}$. Thus, it holds that:

$$P_L \geq \frac{1}{|stream_\mathcal{V}| - n}.$$

Here, n is the length of the prover and verifier nonces, as specified in the protocol. If $\mathcal{A}$ knows the location of $\mathcal{P}$ and $\mathcal{V}$ that implies that $P_p = 1$, while P_q is given by:

$$P_q = \frac{1}{|N_\mathcal{P}|} = \frac{1}{n}.$$

Otherwise, if the adversary does not know the position of the prover and verifier, he must guess the offset p, thus:

$$P_p = \frac{1}{|stream_\mathcal{P}| - |N_\mathcal{P}|} = \frac{1}{|stream_\mathcal{P}| - n}.$$

The success probability if p is known is thus:

$$P_s \geq \frac{1}{n(|stream_\mathcal{V}| - n)}.$$

If p must be guessed, the probability is:

$$P_s \geq \frac{1}{n(|stream_\mathcal{V}| - n)(|stream_\mathcal{P}| - n)}.$$

Significance of this result. In the worst case scenario, that of an adversary who must guess the offset p as well as L and q, the success probability is lower bounded by the value $\frac{1}{n(|stream_\mathcal{V}|-n)(|stream_\mathcal{P}|-n)}$ as shown above. The denominator is polynomial in n, as a minimal efficiency requirement is to demand that the prover and verifier run in polynomial time. To make the protocol usable in practice, both $stream_\mathcal{V}$ and $stream_\mathcal{P}$ must be small. Thus, in fact, the success probability of the adversary is quite large.

Improving the RČ Protocol. In order to combat the attack described in the previous paragraph, we propose the following protocol (depicted in Figure 32).

Security of the Protocol. We only briefly sketch the mafia and distance fraud resistance of our new protocol, assuming that F is a PRF. Our proof works similarly as proofs for round-based protocols, which we showed in Chapter 2.

Theorem 6.6 *Assuming that F is a PRF, that R_V is uniformly distributed in a set of exponential size, that R_P is in a set of exponential size, the protocol in Figure 32 is a distance-bounding protocol which provides resistance to distance fraud and to mafia fraud.*

Proof (Sketch).

Distance fraud. Assuming that a malicious prover $\mathcal{P}$ can authenticate to the verifier $\mathcal{V}$ although he is outside the proximity of $\mathcal{V}$, this prover can be turned into a distinguisher against the pseudorandomness of the string $R_\mathcal{V}$. Indeed, even if F is *not* pseudorandom, authentication requires the correct value $R_\mathcal{V} \oplus R_\mathcal{P}$, where $R_\mathcal{P}$ is part of the outcome of f. Even if the value $R_\mathcal{P}$ is known, the XORed value is masked by the random nonce $R_\mathcal{V}$.

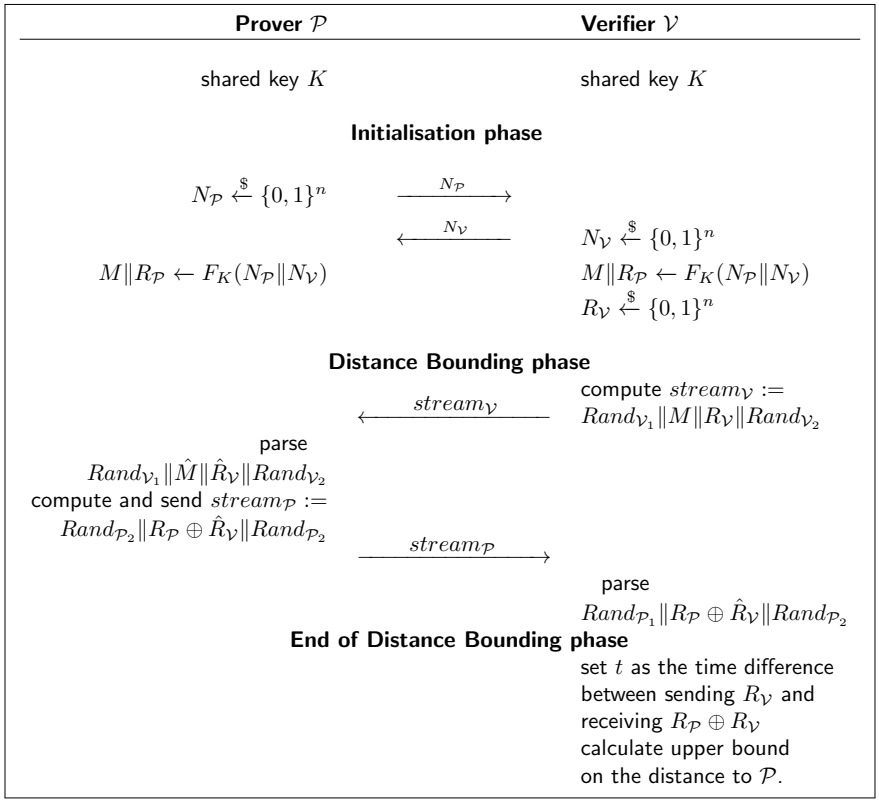

Figure 32: An improved stream-based protocol

Mafia fraud. Let $\mathcal{A}$ be an adversary who runs a mafia fraud between a far away honest prover $\mathcal{P}$ and a verifier $\mathcal{V}$. Following the game reduction technique, we transform the mafia fraud game into a game in which, for all N_P and N_V, there is at most one session of the prover protocol $\mathcal{P}$ and one session of the verifier protocol $\mathcal{V}$ which use N_P and N_V. So, the input to F does not collide in between two prover or two verifier sessions. Then, we replace F by a truly random function in a bridging step based on the PRF assumption on F. We obtain that M and R_P are uniformly distributed for the pair of matching prover-verifier sessions. So, at the critical time when $\mathcal{A}$ must send $R_P \oplus R_V$ (assuming that he correctly guesses the position of M), he received no information about R_P yet from $\mathcal{P}$ who is far away. So, the probability to succeed is negligible. □

6.4.2 Location Private RČ Distance Bounding

We apply our results to our improved variant of the RČ protocol, assuming that the prover and verifier share a secret key K. Our resulting protocol is depicted in Figure 33.
We proceed to also describe the protocol in more detail. The scheme is again composed of two phases: the *initialisation phase* and the *distance bounding phase*.

- **Initialisation Phase:** The prover $\mathcal{P}$ generates a random nonce $N_{\mathcal{P}}$ and sends it to the verifier $\mathcal{V}$The verifier $\mathcal{V}$ generates a random nonce $N_{\mathcal{V}}$ and sends it to the prover $\mathcal{P}$. Both the prover and the verifier use as input the concatenation of the nonces $N_{\mathcal{P}}$ and $N_{\mathcal{V}}$ as input to a keyed pseudorandom function (F_K) and divide the output of the prf into two parts, i.e.: $M \| R_{\mathcal{P}} \leftarrow f_K(N_{\mathcal{P}} \| N_{\mathcal{V}})$. Finally the verifier $\mathcal{V}$ generates another random value $R_{\mathcal{V}}$ of length n.
- **Distance Bounding (DB) Phase:** Distance bounding proceeds as follows:
  - The prover $\mathcal{P}$ waits for a delay Δ that follows the uniform distribution with range $[0, B]$, where B satisfies the

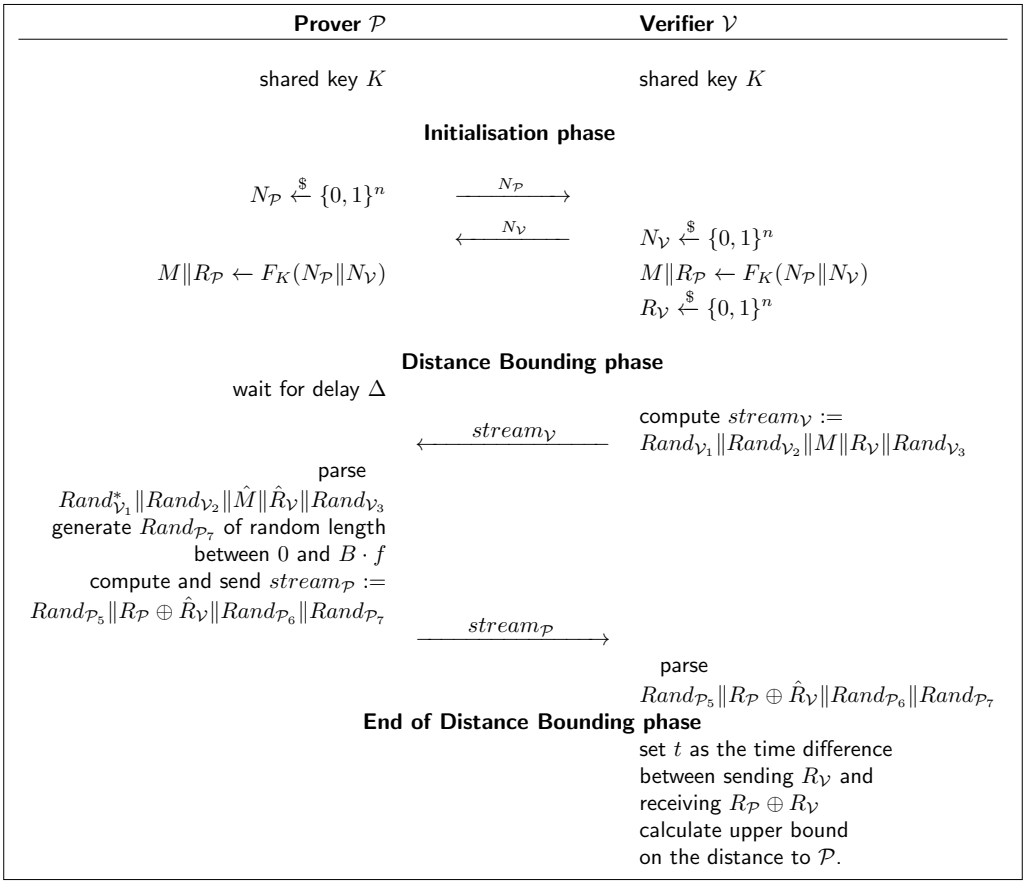

Figure 33: A location-private distance-bounding protocol

following condition (as explained above):

$$B \geq 2^{h+1} t_{\max}$$

After waiting for this delay Δ if the prover $\mathcal{P}$ has still not received any bits from the verifier $\mathcal{V}$ it starts transmitting random bits $Rand_{\mathcal{P}_1}$. In total, if the maximal transmission frequency is f, the length of $Rand_{\mathcal{P}_1}$ is $\max((\Delta\mathsf{t}(\mathcal{P},\mathcal{V}) - \Delta) \cdot f, 0)$.

– While the prover $\mathcal{P}$ waits for a delay Δ (see above), the verifier $\mathcal{V}$ transmits a continuous stream ($stream_{\mathcal{V}}$) such that:

$$stream_{\mathcal{V}} := Rand_{\mathcal{V}_1}\|Rand_{\mathcal{V}_2}\|M\|R_{\mathcal{V}}\|Rand_{\mathcal{V}_3}$$

Here, the length of $Rand_{\mathcal{V}_1}$ is exactly $B \cdot f$ bits, where f is the transmission frequency of the verifier (i.e. the number of bits that it transmits during 1 time unit). The lengths of $Rand_{\mathcal{V}_2}$ and $Rand_{\mathcal{V}_3}$ are chosen uniformly in the range $[0, R]$ [13]. We denote by $L = |Rand_{\mathcal{V}_1}| + |Rand_{\mathcal{V}_2}| + |Rand_{\mathcal{V}_3}|$ the total length of the randomness sent by the verifier. Then, the length of the verifier's stream is exactly $L + |M| + |N^1_{\mathcal{V}}|$, where $|M|$ is the length of the hidden marker and $|R_{\mathcal{V}}|$ is the length of the verifier's nonce $R_{\mathcal{V}}$. The values $|M|$ and $|R_{\mathcal{V}}|$ are global invariants for the protocol, whereas L is session specific. Thus, the hidden marker M is sent at a position $Bf + |Rand_{\mathcal{V}_2}|$.

[13] In [69], it is stated that having $|Rand_{\mathcal{V}_2}|$ and $|Rand_{\mathcal{V}_3}|$ random prevents the adversary from guessing the location of the marker M. However, it is not clear that it would translate into a privacy leakage in our variant due to [58]. It seems that choosing $R = 0$ would work. We let this analysis as an open problem.

– As soon as $\mathcal{P}$ has waited its random delay, it parses the received values. In general, we denote by $\text{Shift}_p(s)$ the string s shifted by an offset p. If p is positive, it means we drop the p leading bits of s. Otherwise, it means that we prepend p zero bits to s. Then, $\mathcal{P}$ parses the received stream as: $\text{Shift}_{-p}(Rand_{\mathcal{V}_1})\|Rand_{\mathcal{V}_2}\|\hat{M}||\hat{R_{\mathcal{V}}}||Rand_{\mathcal{V}_3}$, where $p = (\Delta\mathsf{t}(P,V) - \Delta) \cdot f$. In other words, the shift accounts for the waiting time of Δ time units and the distance it takes for each bit of the stream to get from $\mathcal{V}$ to $\mathcal{P}$. We denote $\text{Shift}_{-p}(Rand_{\mathcal{V}_1}) = Rand^*_{\mathcal{V}_1}$. As the prover parses the received bits, it computes and sends the stream ($stream_{\mathcal{P}}$) such that:

$$\begin{aligned} stream_{\mathcal{P}} &:= [(Rand_{\mathcal{P}_1}||Rand_{\mathcal{P}_2}) \oplus (Rand^*_{\mathcal{V}_1}\|Rand_{\mathcal{V}_2})]\|(Rand_{\mathcal{P}_3} \oplus \hat{M})\|\hat{R}_{\mathcal{V}} \oplus \hat{R}_{\mathcal{P}}\| \\ &\quad (Rand_{\mathcal{P}_4} \oplus Rand_{\mathcal{V}_3})\|Rand_{\mathcal{P}_7} \\ &= Rand_{\mathcal{P}_5}\|R_{\mathcal{P}} \oplus \hat{R}_{\mathcal{V}}\|Rand_{\mathcal{P}_6}\|Rand_{\mathcal{P}_7} \end{aligned}$$

where we denote:
$Rand_{\mathcal{P}_5}$:=$[(Rand_{\mathcal{P}_1}||Rand_{\mathcal{P}_2}) \oplus (Rand^*_{\mathcal{V}_1}\|Rand_{\mathcal{V}_2})]\|(Rand_{\mathcal{P}_3} \oplus \hat{M})$ and $Rand_{\mathcal{P}_6} := Rand_{\mathcal{P}_4} \oplus Rand_{\mathcal{V}_3}$.
Note that $Rand_{\mathcal{P}_7}$ has a random length between 0 and $B \cdot f$ to defeat location-privacy loss based on the analysis of the last stream bit reception.

– After the end of the *distance bounding phase* the verifier $\mathcal{V}$ calculates the time difference t between sending $R_{\mathcal{V}}$ and receiving $R_{\mathcal{V}} \oplus R_{\mathcal{P}}$. The verifier authenticates the prover $\mathcal{P}$ iff. $t \leq t_{\max}$.

7 Significance and Impact of our Results

In this chapter we aim to put our results into a wider perspective. In particular, coming back to the motivation of having clear and formal security models for distance-bounding protocols, we explain in how far our definitions achieve this. In view of our results, security in distance-bounding protocols appears to be a much broader topic than initially thought of, as indeed our refinements of mafia and terrorist fraud resistance clearly show. We furthermore stress the idea that particular care should be taken to the distance-bounding scenario with multiple provers and multiple verifiers, which is an immediate direction for further work.

Before discussing the impact of our results in more detail, we first give in section 7.1 an overview of our contributions, matching them against previous or concurrent work. We divide our results into four categories: (1) Aspects of Modelling; (2) Exact Protocol Assessment and Security Breaches; (3) Tools; and (4) Constructions.

In the first category we include the notions we defined throughout the paper, from the initial framework of Chapter 2 to the notions of privacy (both authentication and location privacy), and to the refinements to mafia and terrorist fraud resistance in Chapters 4 and 5. We try to give here an overview of the various notions we introduced and how they relate to each other, also comparing them to previous or parallel results. Under the second category, we summarise our results and conclusions regarding various distance-bounding protocols in the literature: Brands and Chaum [13], Hancke and Kuhn [42], Avoine and Tchamkerten [6], Kim and Avoine [49], Bussard and Bagga [15], Reid et al. [70], and the Swiss-Knife Protocol [50]. Under attacks, we list our mafia fraud attack against the Rasmussen-Čapkun distance-bounding protocol, as well as key-learning attacks and the generic attack we give against the terrorist fraud resistance of the Bussard-Bagga and the Reid et al. schemes. Recall also that a direct consequence of the results of Boureanu et al. [11] is that most distance-bounding protocols in the literature are in fact not distance-fraud resistant. We also note that both these protocols attain an independent notion of privacy, i.e. GameTF security. We discuss in more detail the impact of this apparent contradiction and its consequences on general terrorist fraud resistant constructions. Furthermore, we note that key-learning attacks —which are very well known in the context of authentication— are usually not considered in distance bounding, although they seem easy to implement.

Categories (3) and (4), i.e., Tools and Constructions, are somewhat related, though not identical. Under tools we give mostly generic tools presented in this thesis, such as the key update compiler that adds narrow-destructive privacy to distance-bounding protocols. A further tool is the compiler presented in Chapter 4, which is discussed in the perspective of the property it achieves, namely strong mafia fraud resistance. Under Constructions, by contrast, we list the particular schemes we outline throughout this thesis, with particular emphasis on the terrorist fraud resistance scheme in Chapter 5 and the location private, enhanced RČ protocol of Chapter 6. We do not count our improvement to the protocol due to Kim and Avoine, which we present in Chapter 2 as a construction, but rather list these modifications under the protocol analysis section.

Following the overview of our contributions we more extensively discuss, in Section 7.2, the impact of our results on present-day and future distance-bounding constructions and models. In particular, we give a classification of security notions, indicating which notions should be achieved by various practical deployment scenarios, such as public transport or personal identification. By relating this back to the protocol analysis we do throughout the thesis, we are able to indicate which distance-bounding protocols may be suitable in several scenarios. Finally we also discuss the impact of our models on distance bounding as a discipline, showing that the attacks we present serve as a motivation to encourage precise formalisation of security notions. We also briefly discuss the benefits of exact (as opposed) to asymptotic security statements.

Finally, we conclude this chapter with section 7.3, where we discuss directions for future work. Here we identify three main research directions, which can be seen as natural follow-ups of our research here: (1) Modelling, where we mention in particular the extension of the channel-based model in Chapter 6, and the treatment of multiple-prover-multiple-verifier distance-bounding scenarios; (2) Constructions, where an important next step would be to consider protocol design with respect to recent advances in the area of Elliptic-Curve processors for RFID hardware, which enables EC computations at a lower cost than PRF (HMAC) implementations; of further motivation in this respect are recent discussions about the practical security of HMACs [51], which indicate that in HMACs are not as secure in practice as they are assumed to be: thus, for RFID hardware it may be infeasible to implement secure HMACs; (3) Implementations, which would feature a comparison of the implemented schemes in the literature, followed by an analysis of the feasibility of deploying distance-bounding in practical scenarios.

7.1 Overview of Contributions

7.1.1 Aspects of Modelling

In Chapter 2, we introduce a framework that models what is usually understood by security in distance-bounding protocols. As in the previous framework due to Avoine et al. [4], we suggest the fundamental four notions in distance bounding are: (1) mafia fraud resistance; (2) terrorist fraud resistance, which we call SimTF security, as in Chapter 5; (3) distance fraud resistance; and (4) impersonation security. Furthermore, in Chapter 3, we introduce the notion of (5) availability, and reiterate the notion of (6) privacy, introduced in the context of RFID by Vaudenay [73]: these two notions are given in the context of key updates, thus adding an extra dimension to the basic security model. A final basic dimension is given in Chapter 6, where we define (7) *location* privacy for phase-less protocols. We depict this general picture in Figure 34.

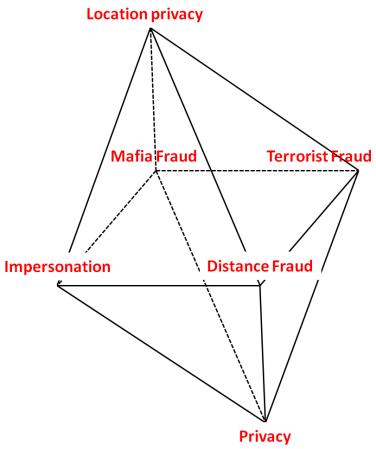

Figure 34: Model in a nutshell

We also show that the basic model we presented in Chapter 2, whereas it may be minimally sufficient for the design of distance-bounding protocols, is not in fact sufficient to cover all possible attacks. In particular, we show in Chapter 4 that key-learning attacks can easily be implemented in practice, especially against protocols aiming to achieve SimTF security, since in such protocols, the prover's responses are related to each other and in turn related to a long-term secret key. This motivates our definition of (8) KLMF security, which is a variation of mafia fraud resistance where the adversary may additionally learn information about the secret key by interacting with the prover and the verifier during observation sessions. Further exploring the idea of aborts in this setting, we define (9) strMF security, i.e. strong mafia fraud resistance. Our work in Chapter 5 has a slightly different motivation, seeking in fact a minimalistic (and practical) version of terrorist fraud resistance, which enables to both have efficient constructions and to capture the intuition of several existing protocols and of the framework of Avoine et al. [4]. This notion is (10) GameTF security. Furthermore, we seek to extend the SimTF notion which we gave in Chapter 5 to capture the strongest possible adversarial attacks, with the motivation that such adversarial power enables high-security constructions. This strong notion is (11) strSimTF security.
We compare these notions in Figure 35, and describe them shortly in the following:

1. **Mafia Fraud.** This is a typical Man-in-the-Middle (MITM) attack, where the adversary attempts to impersonate a legitimate prover while this prover is far away. The adversary can relay lazy-phase messages between the prover and the verifier, but not time-critical messages, which should be either modified as they are relayed, or they should be sent in a non-relaying scheduling.
2. SimTF **Security.** In this scenario, the adversary is aided (in a strictly offline manner) by a dishonest prover in its impersonation attempt. The attack is successful if the adversary authenticates with some probability p and it is impossible to recover enough information from the interaction between the adversary, the dishonest prover, and the verifier, in order to later authenticate *with equal probability* p.
3. **Distance Fraud.** This attack is mounted by a dishonest prover, placed outside the verifier's proximity. The goal of the adversary is to authenticate, thus fooling the verifier into believing the prover is within the prescribed proximity.
4. **Impersonation Attacks.** This attack is motivated by the difficulty of resource constrained devices to sustain many rounds of communication. Since the overall security of the protocol depends on the number of time-critical rounds,

resource constrained devices have an overall low security degree for a small number of time-critical rounds. We thus strongly suggest that distance-bounding protocols should necessarily also contain some lazy-phase authentication. In impersonation attacks, adversaries attempt to authenticate during the lazy phases of a protocol.

5. **Availability.** This notion is an extension of completeness in the presence of key updates. An adversary against availability tries to desynchronise the prover from the verifier, thus introducing Denial-of-Service (DoS).
6. **Privacy.** The scenario of privacy was previously defined in the scenario of RFID authentication. We use the notion due to Vaudenay [73], where the adversary interacts with provers (associated with virtual handles) and with the verifier, and may corrupt provers. The ultimate goal of the adversary is to trace provers better than a so-called blinded adversary, for which a simulator (called a blinder) simulates all but the corruption queries.
7. **Location Privacy.** The adversary here must distinguish between multiple copies of the same prover, placed at different distances from a verifier. The adversary is aware of the verifier's location, and learns all the messages exchanged by the prover and the verifier, together with the times at which these messages arrive at the adversary. The adversary also knows the sending time of messages that originate at the verifier. Stronger adversaries are distributed, consisting of two adversarial entities, and also know the sending time of messages that originate at the prover.
8. KLMF **Security.** In this scenario, the adversary interacts in a regular MITM mafia interaction with the prover and verifier, but may also intervene in the communication of honest provers and verifiers, thus learning key-related information. The adversary then has to authenticate to the verifier, in the presence of a distant honest prover.
9. strMF **Security.** This model extends KLMF security in the sense that the adversary may also "take over" from the honest prover and authenticate in a session between it and the verifier. The main difference is thus that now the adversary can also authenticate while the prover is within proximity and trying to authenticate. We thus model security in the presence of aborts.
10. GameTF **Security.** Here the adversary receives aid from a dishonest prover (both during lazy and during time-critical phases), in order to impersonate the prover to the verifier. The attack is valid if ulteriorly an adversary sharing state with this GameTF adversary cannot win in a mafia MITM communication with the prover and the verifier.
11. strSimTF **Security.** Finally this is an extension of the SimTF notion, where the adversary also receives aid from the dishonest prover during time-critical session, with the restriction that no relaying scheduling is used.

Notion	Attack Goal	Game vs. Simulation	Target Phase	Static Key	Key Learning
Mafia Fraud	impersonation	Game	Time-Critical	✓	×
SimTF **Security**	impersonation[1]	Simulator	Time-Critical	✓	×
Distance Fraud	distance fraud	Game	Time-Critical	✓	×
Impersonation	impersonation	Game	Lazy	✓	×
Availability	Denial of Service (DoS)	Game	All phases	×	×
Privacy	traceability	Simulator (Blinder)	All phases	×	×
Location privacy	location distinguishability	Game	None[2]	✓	×
KLMF **Security**	impersonation	Game	Time-Critical	✓[3]	✓
strMF **Security**	impersonation for aborts	Game	Time-Critical	✓[3]	✓
GameTF **Security**	impersonation[4]	Game	Time-Critical	✓	×
strSimTF **Security**	impersonation[1]	Simulator	Time-Critical	✓	×

Figure 35: Distance-Bounding Notions at a Glance. Notes: [1] The impersonation here is done with the prover's help, and the success is matched against a simulator. [2] Location privacy is defined for phase-less protocols. [3] Though we achieve these notions by applying some form of key update, the notions of strMF and KLMF security do not require it. [4] Here, the impersonation is matched against an ulterior mafia fraud attack

Independence of Notions. A very important modelling aspect of our framework is to show the independence of the four standard security notions that apply to distance-bounding protocols, i.e. mafia, terrorist, distance, and impersonation security. In particular, in Chapter 2 we show as separations distance-bounding protocols which have three of the four properties but not the fourth. Though we give more details on the tools we use in our proofs in Section 7.1.3, we review in this notion the essentials of each of these four notions.
An essential aspect of mafia fraud resistance is the restriction of tainted phases. In particular, our mafia fraud model is very permissive to the adversary, as a time-critical phase is *not* tained if the adversary either (a) flips the bits transmitted between the honest prover and the honest verifier; or (b) schedules messages other than in relay scheduling. Thus, an adversary can win if flipping either the challenge or the response bits during sessions is an effective strategy to authenticate. Additionally, the Go-Early strategy, where the adversary first queries the prover (trying to guess the verifier's challenge) and then receives the genuine challenge is also a powerful tool towards breaking mafia fraud resistance (as will be further seen in section 7.1.2).
The essence of achieving terrorist fraud resistance is to give the simulator an additional advantage, compensating for the erroneous and tainted phases. The key point in our model is the existence of the simulator, who (a) begins its own attack only after the terrorist adversary, aided by a dishonest prover, has succeeded in an authentication attempt; (b) gets access to the full adversary view, including its internal randomness (thus, in particular, it is able to rewind the adversary); (c) must run at most the same number of sessions with the verifier as the adversary does, and must win with the same probability as this adversary. The essence of our definition is that a protocol is terrorist fraud resistant if for every adversary there exists such a simulator, which wins with the same probability as the adversary. In particular, the intuition behind this notion is that in a terrorist fraud resistant distance-bounding protocol, any help the prover gives to the adversary either (i) is ineffective towards making the adversary authenticate; or (ii) enables sufficient information to be recovered for all subsequent attacks to be as effective as the original adversary authentication.
Distance fraud resistance is not implied by either mafia or terrorist fraud resistance because it is essentially a different type of attack. In terrorist fraud resistance, the terrorist adversary is within the verifier's proximity, but only has limited offline support from the dishonest prover. In the distance fraud model, the adversary *is* the dishonest prover, but is *not* in proximity. Thus, distance fraud captures a prover's ability to make the verifier believe the prover is within proximity, whereas the goal of terrorist fraud is to enable an adversary (within proximity) to authenticate without having complete prover information. Thus we both intuitively and formally refute previous statements to the effect that terrorist fraud resistance implies distance fraud resistance.
Finally, impersonation security refers to lazy phase authentication only. The model here captures an attack where the adversary tries to authenticate to the verifier, strictly without relaying messages. Note that in the mafia fraud scenario the MITM adversary is allowed to relay message in lazy phases (since the clock is only employed in time-critical rounds). However, impersonation security only captures the typical impersonation security scenario in authentication. This notion is also independent of the previous ones, since it only affects lazy phases, and not time-critical ones.

We further recall the relationship between mafia and terrorist fraud, with their corresponding sub-flavours. The connecting point here is that mafia and strSimTF security imply GameTF security, whereas strSimTF security is independent from GameTF security. We depict these relationships in Figure 36.

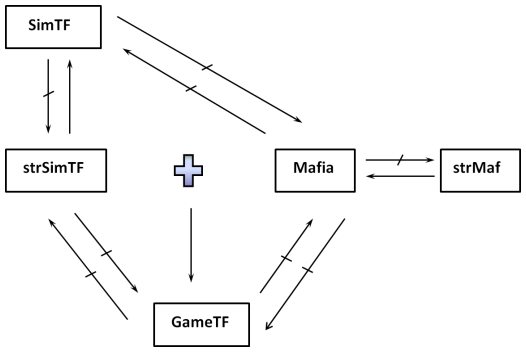

Figure 36: The mafia and terrorist fraud flavours

The Need for Privacy. Though privacy is a rising concern in authentication, very few protocols in the distance-bounding literature consider this idea. In fact, the only construction aiming to achieve privacy is the Swiss-Knife protocol of Kim et al. [50], whereas Rasmussen en Čapkun introduce the (informally sketched) idea of location privacy. However, to our knowledge, our work is the first to formalise the notion of location privacy, and to formally define availability as long-term completeness for stateful distance-bounding protocols. Whereas such notions are well-known in the context of authentication, they were never defined in the context of distance-bounding.
So is privacy really necessary in distance-bounding? We answer in the affirmative, noting that distance-bounding is simply an enhancement of regular authentication, meant to prevent an additional attack (MITM relaying). We argue that distance-bounding should replace regular authentication, at least in environments where relay attacks have actually been implemented and they have proved effective. Since application scenarios for authentication range from personal identification to logistics and public transport, we suggest that privacy is an essential property for distance-bounding protocols. As a consequence of our results, we also have an easy transformation from regular distance-bounding to stateful distance-bounding protocols (i.e. with key updates) see Chapter 3 and the following section 7.1.3, which gives an overview of Tools. We also argue that, in environments where computational location privacy can be achieved without too great a loss of efficiency (in view of our results in Chapter 6), our proposed countermeasure of introducing a delay in communication should be implemented.

7.1.2 Protocol Assessment and Security Breaches

In Chapter 2 we review and assess the security properties of a number of protocols, namely: Brands and Chaum [13], Hancke and Kuhn [42], Avoine and Tchamkerten [6], Kim and Avoine [49], Reid et al. [70], and the Swiss-Knife Protocol [50]. These protocols are analysed in our basic framework, also presented in Chapter 2. Furthermore, in Chapter 4 we review the security of the protocol due to Bussard and Bagga [15], in the same model. Before outlining the results of Chapters 4 and 5, where we describe further attacks, we summarise these results in Figure 37.

	Mafia	SimTF	**Distance**	**Impersonation**
[13] [1]	$\left(\frac{1}{2}\right)^{N_c}$	×	×	$\left(\frac{1}{2}\right)^{N_c}$
[42]	$\left(\frac{3}{4}\right)^{N_c}$	×	×	×
[6] [2]	$\frac{1}{2}(N_c+2)\left(\frac{1}{2}\right)^{N_c}$	×	×	$\left(\frac{1}{2}\right)^{\lvert V\rvert}$
[49]	$\left(\frac{1}{2}\right)^{N_c}$	×	×	×
[15]	$\left(\frac{3}{4}\right)^{N_c}$	×	$(\left(\frac{3}{4}\right)^{N_c})^3$	×
[70]	$\left(\frac{3}{4}\right)^{N_c}$	×	$(\left(\frac{3}{4}\right)^{N_c})^3$	×
[50]	$\left(\frac{1}{2}\right)^{N_c-T}$	×	$\left(\frac{3}{4}\right)^{N_c-T}$	$\left(\frac{1}{2}\right)^{\lvert V\rvert}$
[27]	$\left(\frac{1}{2}\right)^{N_c}$	×	×	$\left(\frac{1}{2}\right)^{\lvert V\rvert}$

Figure 37: Distance Bounding at a glance. [1]This protocol uses expensive primitives. [2]This protocol requires exponential storage requirements. ^{*3} This property depends on an additional assumption, namely that an additional secret key is chosen honestly and at random. We denote by N_c the number of time-critical rounds, by T a tolerance level for faults, and by $|V|$ is the bit length of an authentication string V sent by the verifier

Apart from enabling easy protocol comparison in our formal framework, our analysis also highlights a few important issues, which affect both protocol design and protocol assessment. Most of these points have already been discussed in Chapter 2, but we reiterate some of the more essential ideas here.

The Go-Early Strategy. Firstly, we note that in mafia fraud attacks, the so-called Go-Early strategy, where the adversary first queries the prover in a separate session about time-critical responses, and then responds to the verifier in the authentication session, is usually the most effect time-critical strategy that adversaries can employ. This strategy works very well in protocols where the time-critical challenges sent by the verifier are independent of each other and not somewhat tied to a latter transcript authentication. The consequence of this attack can be seen in the rate of success of mafia fraud adversaries in Figure 37. In particular, protocols like the Hancke and Kuhn protocol are *insecure* against this strategy, whereas protocols where the challenges are inter-related, e.g., the Avoine-Tchamkerten and resp. the Kim-Avoine protocol, are better protected against this strategy. However, we note that the Avoine-Tchamkerten construction achieves this property at the expense of storage, whereas the Kim-Avoine protocol has to compromise in distance fraud resistance.

A very elegant way to protect against the Go-Early strategy is in fact to authenticate the time-critical transcript of the protocol, as it is done in the Swiss-Knife protocol, following an idea by Brands and Chaum.
We note that the direct consequence of the success of the Go-Early strategy is that we require provers and verifiers to run more time-critical phases for e.g. the Hancke-Kuhn protocol than for a scheme like the Swiss-Knife protocol. In this context, we emphasise the advantage of *exact* versus *asymptotic* security. Both the Hancke-Kuhn and the Swiss-Knife protocol are asymptotically mafia fraud resistant: however, a particular choice of N_c may make the Swiss-Knife protocol secure in practice while enabling a computationally feasible attack against the Hancke-Kuhn protocol. Thus, by giving exact security statements we are also able to compare the efficiency with which the protocols attain a particular property. This is also the case for the other properties, in particular distance fraud resistance, where we can see that the Kim-Avoine protocol is deficient with respect to other constructions.

SimTF **Security and Attacks.** Secondly, it seems that our original notion of terrorist fraud resistance, denoted SimTF security, is in fact quite strong. In general, protocols attempting to achieve terrorist fraud resistance attempt to inter-relate the challenges used during time-critical phases such that if an adversary knows both possible responses for a time-critical phase, it recovers a bit of the secret key. This is the case for protocols like Bussard-Bagga and Reid et al. The intuition is that such schemes prevent terrorist fraud attacks because if the adversary receives both response bits from a dishonest prover, it has an advantage in further authentication sessions. However, the SimTF definition states that an attack is trivial only if the simulator, given access only to the adversary's view, is able to authenticate with the same probability: else, the attack is valid and the scheme is *not* terrorist fraud resistant. Our terrorist fraud attack against both the Bussard-Bagga and the Reid et al. schemes exploits precisely this semantic difference.
The dishonest prover in our attack gives the adversary the correct responses for each time-critical round for which the challenge is 0. This response string reveals no information about the long-term secret. For rounds where the challenge is 1, the adversary must guess the correct response, and does so with probability $\frac{1}{2}$ per round. Thus, the adversary's overall success probability is about $\left(\frac{3}{4}\right)^{N_c - T_{\max} - E_{\max}}$, where N_c denotes the number of time-critical rounds. The simulator starts its authentication attempt after the adversary authenticates successfully, thus it may be able to infer a number of bits of the secret key, equalling the number of rounds of the successful authentication session where the adversary correctly guessed the response for the rounds with 1 challenges. However, this will not help the simulator authenticate, since it knows nothing about the freshly computed response: indeed, the simulator's only hope is to guess the remaining bits of the secret key and thus impersonate the prover, for which it has a smaller probability to succeed than the adversary. Furthermore, note that the adversary can also taint rounds, which the simulator cannot. We discuss the intuitive degree of terrorist fraud resistance that the Bussard-Bagga and the Reid et al. protocols achieve in a later paragraph of this section, when we review the flavours of terrorist fraud resistance.

Mutual Authentication and Distance Fraud. As previously mentioned, the protocol due to Kim and Avoine achieves better resistance to the Go-Early mafia fraud strategy by using mutual authentication. In particular in this protocol the verifier also authenticates during some of the time-critical phases, such that the prover can verify if the received challenge is the expected one (else, it aborts). However, we note that this method enables a distance fraud adversary to predict some of the challenges that the verifier sends. Thus it is able to fool the verifier's clock for these phases. An optimal distance fraud resistance is achieved by simply asking the prover to echo the challenges it receives; however, this enables mafia and terrorist fraud attacks.

We also discuss in this thesis various flavours of mafia and terrorist fraud resistance. In particular, we show that the intuitive degree of terrorist fraud resistance attained by the protocols due to Bussard and Bagga (and by extension the one due to Reid et al., which is very similar) is captured by our definition of GameTF security.

Insight: What GameTF **Security Means.** The GameTF definition requires that once a strSimTF fraud adversary, succeeding with non-negligible probability in its attack, passes on its state to another adversary, this second adversary authenticates, using a mafia-fraud interaction, only with at most mafia-fraud resistance. Intuitively, this is equivalent with requiring that the state passed between the adversaries is worthless. This model tries to capture the two-faceted nature of a terrorist fraud attack. Here, the terrorist adversary wants to not only authenticate once with the dishonest prover's help, but also to be able to use the information given by the prover to authenticate afterwards. Thus, if the dishonest prover aids the adversary in this endeavour, we call the attack invalid.
This formalisation restricts the power of a terrorist attack quite a lot, because the prover can only forward non-essential information; however, this constraint seems to capture the intuition behind terrorist fraud attacks as defined e.g. in [4].

Such a restricted adversary in particular cannot mount the generic attack we showed for terrorist fraud resistance, and also it cannot mount any attack where the terrorist adversary's success is attained by means of revealing some significant information, such as bits of the secret key, as seen in the case of the Bussard and Bagga protocol.

SimTF **and** strSimTF **Security.** Is GameTF security sufficient? We argue that whereas GameTF security is the form of terrorist fraud resistance most frequently addressed and met by the literature, stronger security requirements are needed in e.g. scenarios where the result of successfully authenticating are so great that the dishonest prover may be willing to sacrifice some partial information on the secret key. This could be the scenario where successful authentication allows an adversary to bypass border control or customs (e.g. for traffic of counterfeit drugs). In particular, SimTF and strSimTF security place a much weaker restriction on the validity of an attack. In particular, an attack is invalid in these setting only if enough information can be extracted (by a simulator) from the terrorist adversary to authenticate *with the same probability*.
We also show that these strong requirements can, actually, be achieved, as we show the first SimTF and strSimTF fraud resistant scheme in the literature. Note that as strSimTF security implies SimTF security, the known protocols in the literature are also not strSimTF secure. The notion of strSimTF security gives the adversary more power, by enabling it to communicate with the dishonest prover during time-critical rounds, with the restriction that no relaying scheduling may be used. Thus, in constructions we need to ensure that the time-critical information forwarded by the verifier can only be relayed to the prover: in particular, the challenges should not be predictable for any phases.
We summarise our assessment of the various flavours of terrorist fraud resistance in Figure 38.

	SimTF	strSimTF	GameTF
[15]	×	×	✓
[70]	×	×	✓
Our Scheme	✓	✓	✓

Figure 38: Terrorist fraud construction at a glance

Mafia Fraud and the RČ Protocol. We also show an individual mafia fraud attack against the protocol due to Rasmussen and Čapkun in Chapter 6. In this protocol, the prover and verifier use both an encryption scheme and a digital signature scheme, exchanging two encrypted and signed messages in an initialisation phase. The most important note here is that the prover nonce can be replayed across sections, such that the authentication is run directly on this nonce, and not on fresh values, which are session dependent. After the initialisation phase, the prover and verifier communicate in constant, parallel bit streams: after a randomly chosen delay, the verifier will send a shared ephemeral session-specific secret value called a hidden marker. Upon receiving this value, the prover XORs the following bits sent by the verifier with the same nonce that the prover sent in the initialisation phase. Our attack exploits the replayability of this nonce. In particular, the adversary first runs a MITM attack between an honest prover and an honest verifier, from which it extracts two pieces of information: (1) the encrypted and signed nonce which is sent as the prover's first message; (2) by guessing the offset after which the verifier sends the hidden marker (whose length the adversary knows), the adversary can see the value of the reader's nonce, and the reader's nonce XORed with the prover's nonce, from which the adversary can guess the prover nonce used in this session. Thus, at the end of a successful guess, the adversary learns both the nonce, and the encrypted and signed value of it. Now the adversary repeats the following attack for a polynomial number of times: in a session with the verifier, the adversary forwards first the encrypted and signed replayed nonce. Then, during the distance-bounding phase, the adversary guesses when the hidden marker has been sent and then XORs the bits it receives in constant stream with the guessed value of the nonce. The repetition increases the adversary's ability to guess where the hidden marker will be sent in the authentication session, thus increasing its success probability.
The success of this attack is non-negligible, and even significant when the protocol is communication-efficient, that is to say the prover and the verifier do not use very long delays. In case the lengths of the hidden marker and the nonces chosen by the prover, resp. verifier are very large in comparison with the verifier's delay, this probability is even greater.

Achieving Location Privacy. Our analysis shows that provers running distance-bounding protocols always leak more information about their location than just the fact that they are within the verifier's proximity. However, we also show that protocols can attain computational location privacy against limited adversaries (these are single — as opposed to distributed— adversaries who are unable to learn the times when provers send their messages) at the expense of a delay

exponential in the number of bits of location privacy bits that is desired. Whereas this is in theory unacceptable, in practice the high transmission speed of radio-based transmissions allows this delay to remain reasonably low. We note in particular that frequency hopping is not sufficient to effectively hide communication being exchanged by two parties, since in practice the chosen frequencies can *all* be eavesdropped, and the communication can thus be pieced together. In fact, our location-privacy attack in Chapter 6, where we prove the impossibility of achieving information-theoretic location privacy, relies only on eavesdropping the first bit of communication, which an adversary *does* in practice achieve with very great probability even in the context of frequency hopping.

Key-Learning Attacks. In Chapter 2 we show a first key-learning attack against the protocol due to Reid et al. We further discuss such attacks in more detail in Chapter 4. Key-learning attacks are particularly applicable to GameTF secure protocols in the literature. In order to achieve GameTF resistance, the provers relate the two response strings for the time-critical phases by means of a long-term secret key, such that revealing the corresponding bits of both response strings during a time-critical phase reveals a bit of the long-term secret.
This enables an attack that is well-known for authentication scenarios, but which has never been considered for distance-bounding, which we call key-learning. In this key-learning attack, instead of simply observing the interaction between an honest prover and an honest verifier, the adversary observes *most* of the communication, but interferes in a small number of time-critical phases, flipping the challenge bits in order to learn the corresponding bits of the secret key. This attack is not covered by the distance-bounding model of Avoine et al. [4], but it *is* covered in our framework, since we allow arbitrary adversary strategies. In particular, in the mafia fraud model, we only assume that the prover must be out of the verifier's proximity in its final authentication attempt; in this session, the adversary runs a MITM attack in which it may relay at most *a few* phases of communication. However, in previous sessions, the adversary could have relayed more phases and just interfered in a few rounds.
In practice, key-learning attacks are easy to implement, especially when the prover and verifier communicate over a mobile network, where it is easy to flip transmitted bits. We also note that in most GameTF secure distance-bounding protocols, the length of the long-term secret is only equal to the number of time-critical phases that the protocol runs. If the protocol is furthermore run on resource-constrained devices, this length is not very long, and thus the adversary does not even require many protocol runs between the honest prover and the honest verifier in order to learn the full secret key.
There are two ways to prevent key-learning attacks: one is to prevent leakage of key-related material, which seems difficult to do in protocols aiming to achieve GameTF security, and another is to use key updates in order to ensure that the information leaked by the adversary is in fact useless in a future authentication attempt. We prefer to use the latter approach in order to construct a compiler which transforms a mafia fraud resistant protocol into a KLMF secure protocol.

The Case of Aborts. A stronger flavour of mafia fraud attacks is our strMF concept, where the adversary can "take-over" from an aborting prover during an authentication session. We note that most protocols in the literature are vulnerable to such attacks.

7.1.3 Tools

The Compilers. Throughout this thesis, we introduced several techniques to achieve stronger from weaker security notions: these are the so-called compilers. The first compiler we present is the key update compiler, which takes as input a mafia, distance, and impersonation-secure weakly-private distance-bounding protocol (note that weak privacy is a basic requirement, where the adversary cannot corrupt provers), and outputs a distance-bounding protocol that still preserves the input protocol's properties and furthermore achieves narrow-destructive privacy.
The key update compiler requires an additional essential condition: the key generated by the KGen algorithm should be pseudorandom. Note that the key generation algorithm is usually considered separately from the distance-bounding protocols themselves, i.e. no requirement is made of this algorithm in general. Once instantiated with a key generation algorithm that outputs pseudorandom keys, we note that most of the protocols in the distance-bounding literature fulfill in fact the conditions of the compiler, thus we can apply our results to enhance their weak privacy to narrow-destructive privacy.
Another compiler we present is an algorithm to turn a mafia fraud resistant protocol into a strong mafia fraud resistant protocol. Indeed, our results in Chapter 4 indicate that adding a final authentication phase, where the prover authenticates the transcript of the entire session, will in fact ensure that the adversary can only succeed in authenticating to a verifier — even in the presence of an aborting prover within proximity — with negligible probability. We show in fact that this transformation achieves strong mafia fraud resistance even if the underlying protocol is *not* resistant to key-learning attacks. We depict this in Figure 39, and briefly outline the main technical structure of the compilers below.

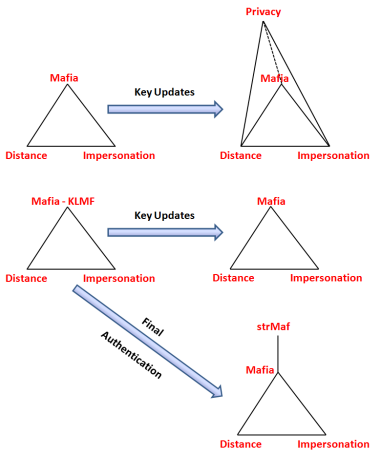

Figure 39: The compilers in this thesis, where the pyramid in the upper level denotes that the resulting protocol achieves privacy apart from the fundamental properties of distance-bounding schemes. In the lower part, (KLMF - Mafia) denotes the fact that the underlying protocol is not resistant to key-learning attacks, but resistant to all other types of mafia fraud.

KEY UPDATES. The key update compiler is a modification of the narrow destructive-private protocol introduced by Vaudenay [73]. There are a few modifications, however. In the protocol used by Vaudenay [73] (which is an authentication, and not a distance-bounding protocol), the verifier never updates state, whereas the prover always updates state (by running a pseudo-random function on the old state). To increase efficiency, we ensure that the verifier also updates state; however, in this context we must ensure that adversaries cannot cause a desynchronisation between the prover and verifier state (a Denial-of-Service – DoS – attack). We achieve this by ensuring that the prover always updates more often than the verifier, and the verifier "catches" up with the prover at the next authentication attempt.

If the underlying protocol does not employ mutual authentication, it is possible to postpone all the verification steps for the distance-bounding phase until the final verification phase; however, for protocols using mutual authentication, the verifier must "catch up" with the prover state before the underlying distance-bounding protocol is run. This is done before the distance-bounding step, by an initial challenge-response lazy phase. Since the response is actually computed by using a pseudo-random function, we can merge this step with the initialisation lazy phases of the underlying protocol if these phases include a PRF-based computation.

strMF. Our strMF compiler adds a final lazy phase prover authentication to the protocol, where the prover sends the output of a PRF computed after the time-critical phases of the protocol are run. This prevents the adversary from taking over from an aborting prover once the prover goes offline, since in that case the adversary has to guess the last lazy-phase authentication response, which it can only do with negligible probability. This compiler also achieves strong mafia fraud resistance if the underlying protocol is not entirely mafia fraud resistant, but rather is vulnerable to key-learning attacks.

Location Privacy. Our main result from Chapter 6 is an impossibility result: we in fact show that location privacy cannot be attained by distance-bounding protocols, unless the parties employ very high delays. However, our proof of this statement also gives us a tool to achieve computational location privacy, if we are willing to use such exponential delays. In fact, the trick here is to ensure that the effective communication time is negligible compared to the delay: in this way we ensure that the adversary cannot guess (except with negligible probability) when the parties will communicate the non-bogus messages. This is a tool for achieving a degree of computational location-privacy that can be employed by distance-bounding protocols. A slight inconvenience is that our model in the location-privacy section is in fact not round-based, thus we cannot easily use these results as a compiler on round-based protocols, such as those outlined in Chapter 2. Intuitively, our results should extend to our original framework, which is round-based, but an interesting direction for future work would be to investigate how far our results extend to this setting.

Small and Handy Tricks. Throughout this document, we also introduce a few small tricks and techniques to either achieve or break several security notions. While not significant enough to count as a construction, these tricks can be handy, either towards achieving a security notion, or towards breaking them. We also show a few proof tricks, which we employ in our proofs throughout this thesis. Though we refer the reader to the main contents of the previous chapters, we also list some of these handy tricks below.

- **Proving mafia fraud resistance.** In Chapter 2 we formally prove the mafia fraud resistance of several distance-bounding protocols. Though the proofs are slightly different for each protocol, the general strategy goes as follows: (1) first we argue that, except with negligible probability, we can replace the response strings computed by the prover by random bit-strings each of length N_c; (2) then we argue that the nonce pairs are unique in a verifier-adversary session, up to a single matching adversary-prover session; and (3) under these assumptions, we argue that the adversary's winning probability is negligible. For the third step, we usually compute the bound recursively, i.e. we look at the probability that the adversary wins $N_c - (i + 1)$ time-critical phases, given that it has already won the first i time-critical phases. We usually analyse the effectiveness of four separate strategies: (i) the Go-Early strategy, which has already been outlined in section 7.1.2, where the adversary tries to guess the verifier's challenge, querying the prover before being queried by the verifier (this ensures the phase is not tainted); (ii) the Go-Late strategy, where the adversary responds to the verifier before receiving the response from the prover (again, this bypasses relaying scheduling, thus the phase is not tainted); (iii) the Modify-It strategy, where the adversary flips the challenge received from the verifier, thus ensuring the phase is not tainted. This strategy is different from the Go-Early strategy: indeed, in the Go-Early strategy the adversary usually has a $\frac{1}{2}$ probability that the adversary guesses the correct challenge; by contrast, in the Modify-It strategy the adversary's challenge is always different from the verifier's challenge; (iv) the Taint-It strategy, where the adversary simply taints the phase and wins (this strategy can only be used in at most $T_{\max}$ phases). A final step in the proof is to bound the success strategy by completing the recursive computation of the success probability.
- **Bypassing mafia fraud resistance.** We use a handy trick in our separation of mafia fraud resistance from distance, terrorist, and impersonation security: we introduce a back door that enables an adversary to break mafia fraud resistance without affecting any other property. We do this by exploiting the definition of tainted phases in mafia fraud. Namely, we modify a protocol which is otherwise mafia fraud resistant such that the prover prepends a bit to its first lazy-phase message. An honest prover always prepends a 0 bit, while an adversary can prepend a 1 bit instead. If it receives a 1 bit, the verifier expects the conjugate bit of the correct response in each of the time-critical phases. This modified protocol is no longer mafia fraud resistant, since the adversary can now bypass the tainted-phase definition and flip the prepended 0 bit forwarded by an honest prover to a 1 bit, thus making the verifier expect the flipped response bits. During time-critical phases, adversary forwards the correct challenges as it receives them from the verifier, then flips the responses that the prover forwards, thus authenticating. However, properties like impersonation security and distance and terrorist fraud resistance (the latter with all its flavours, i.e. SimTF, GameTF, or strSimTF security) are preserved. Intuitively, this holds because in distance fraud, the prover is the dishonest prover itself and thus knows the correct responses (flipped or otherwise, anyway). Impersonation security only affects lazy phases, thus the modification of the protocol is irrelevant to this attack. Finally (though this is a more verbose argument), terrorist fraud resistance is preserved because for every successful adversary, the simulator will be able to use the simulator against the original scheme, removing the prepended bits.
- **Impersonation security.** We have very easily added impersonation security to the protocol due to Kim and Avoine [49], by exploiting the fact that the protocol already uses a pseudo-random function (PRF) in order to compute the prover and verifier output for the time-critical phases. Thus, by employing a PRF with a larger output size, we can include a session-specific authentication string that the prover can forward during authentication phases. This is a trick that can be used in most distance-bounding schemes in the literature, and we argue that it is necessary to make this modification to the protocols which do not offer impersonation security. In particular, since lazy-phase communication is not restricted to a single bit, a single phase of lazy-phase authentication ensures that the total security degree of the scheme is much increased.
- **Mutual authentication.** We note that the protocol due to Kim and Avoine [49] achieves a measure of time-critical mutual authentication, since some of the verifier's challenges are pre-computed and can be verified by the prover. We note that it is also possible to achieve mutual authentication during lazy phases, though we note that the adversary can just forward these messages to the prover during the lazy phases. However, mutual authentication is employed to good use to achieve KLMF and resp. strMF security. As was already explained earlier, in the KLMF compiler the verifier sends, in a lazy phase, a value v, which either authenticates the verifier to the prover (in which case the prover does *not* update its state), or it does not authenticate the verifier (in this case, the prover updates state).

The same trick is employed in our strMF compiler, which builds on the KLMF compiler, adding a last phase of prover-authentication.

- **Proving** GameTF **security.** In Chapter 4 we give a few proofs of GameTF security for various distance-bounding protocols. The structure of the proof is usually as follows: we assume, towards contradiction, that the protocol is *not* GameTF secure. Thus, there must exist a strong terrorist adversary which (1) authenticates (aided online by the dishonest prover) with non-negligible probability, such that (2) *all* adversaries sharing state (the state is the view of the strong terrorist adversary) with the terrorist adversary and running a mafia fraud interaction with the prover and verifier has a probability to authenticate that is upper bounded by the protocol's mafia fraud resistance. The argument usually develops by outlining an adversary as in (2) that wins with greater probability than the mafia fraud resistance of the protocol, which contradicts the assumption and thus proves that the protocol is GameTF secure. For GameTF secure protocols in the literature, we build a second adversary that interacts with the strong terrorist adversary as the simulator does in the SimTF security proofs, but which also uses its mafia MITM interaction with the prover to authenticate (in particular, as opposed to the Simulator in SimTF security, the adversary running a mafia fraud interaction can also taint phases).

7.1.4 Constructions

In this thesis, we have introduced two important constructions: the first SimTF secure protocol in the literature, and a location private protocol based on an improved variant of the RČ distance-bounding protocol due to Rasmussen and Čapkun. We briefly review the main ideas of these protocols, showing how they can be used as general tools in future work.

The SimTF-**Secure Protocol.** This construction, shown in figure 24 in Chapter 5, is the first provably SimTF secure protocol in the literature. We also show that this scheme attains the stronger requirement of strSimTF security. In this construction, we use the standard idea of relating the two response strings for the time-critical rounds by means of a long-term secret key (in fact, the responses are computed such that their bit-wise XOR sum equals a long-term secret; thus, if the long-term secret is chosen honestly and with high entropy, the scheme provides distance-fraud resistance). However, in order to compensate for the adversary's advantage of using the prover's aid, we allow the simulator a back door in its authentication attempt. In particular, the prover prepends a bit to one of its lazy-phase response strings. An honest prover always prepends a 0 bit (this ensures that we preserve mafia fraud resistance and impersonation security); however, if a 1 bit is prepended, the verifier will also accept a guess of the long-term secret key. The probability that the verifier accepts depends on the Hamming distance between the adversary's guess and the real secret key. Denote this distance by d. With this notation, the verifier accepts with probability $2^{-d+T_{\max}+E_{\max}}$, thus allowing the simulator to compensate for the adversary's ability to taint phases or to send erroneous responses.
This back-door for the simulator is essential: if the verifier accepts this guess, then the verifier will flip a flag, and during the time-critical phases it will accept echoed challenges as legitimate responses. Thus, the simulator's strategy is to reconstruct a guess of the secret key that will be accepted with high probability, and during the time-critical phases the simulator can simply echo the challenges, thus authenticating. The accuracy of the simulator's guess depends on the probability that the adversary authenticates successfully, but our construction ensures that if an adversary authenticates with some probability p, our constructed simulator authenticates with the same probability.

The Location Private Protocol. This construction uses ideas from the RČ protocol, but we use two main strategies to modify it. Firstly, we achieve mafia fraud resistance by tweaking the protocol so that the prover's response during the distance-bounding phase (where the clock is employed) is session-specific, thwarting the mafia fraud attack that we show against the RČ distance-bounding protocol. Like in the RČ protocol, we use continuous bitstream transmissions between the prover and verifier, such that the verifier sends a hidden marker, then a random challenge; upon receiving the random marker, the prover continues to transmit, in a constant bitstream, the XOR of the challenge with the ephemeral response computed in the initialisation phase.
The distance-bounding phase is further modified to achieve provable location privacy, by using our results in Chapter 6. In particular, the prover must wait before engaging in the distance-bounding phase for a given delay, which is exponential in the desired number of bits of security. Though this delay is theoretically very large, in practice, for large communication speeds, the introduced delay is not overwhelming in practice.

7.2 Impact of our Results

In the previous section we have reviewed our contributions, dividing them in four separate categories. We briefly discuss the impact of our results in each of these categories on future work and on applications in practice.

Models and Protocols. Our models are meant to serve as a building block for future fundamental work in this area, both for the purpose of assessing existing constructions in the literature, and for a more efficient design of distance-bounding schemes in the future. As argued before, exact, rather than asymptotic security is greatly desirable, making our framework well suited for the assessment and comparison of security properties for distance-bounding protocols. We concretely define six basic, independent notions that are relevant in distance-bounding scenarios: mafia, terrorist, and distance fraud resistance, impersonation security, privacy (in authentication), and location privacy. Contrary to previous results we *prove* that mafia, terrorist, distance, and impersonation security are all independent. Thus, proofs of security for distance-bounding protocols must include separate proofs for all these notions. We also note that our definition of mafia fraud resistance captures an attack that is not previously accounted for by other models, i.e. key-learning attacks.
Furthermore, we showed that a more thorough treatment is required for both terrorist and mafia fraud. In the former case, an acceptable degree of security may be acquired (if the protocol is mafia fraud resistant), with a more restricted adversary: in fact this is the degree of security attained by most protocols in the literature. However, terrorist fraud resistance is a very strong attack, and scenarios where successful authentication results in passing border control or customs require a stronger model, where the adversary is less restricted. We also provide corresponding models, and a construction that attains our stronger security requirements. In practice, our scheme can be used for stronger security requirements, whereas the existing protocols in the literature are proved to suffice for scenarios where dishonest provers are reluctant to yield *any* information that would help an adversary authenticate later (such as e.g. a few bits of the secret key).
For the latter case of mafia fraud resistance we outline an attack that was not covered by previous distance-bounding models [4]. This so-called key-learning attack allows an adversary to learn information about the key shared by a prover and a verifier by interfering in a session run between the prover and the verifier. We stress that this attack is very easily implemented and particularly effective against protocols aiming to achieve some degree of terrorist fraud resistance. Whereas the previous model of Avoine et al. does *not* capture this attack, our framework does allow for key-learning adversaries, and we in fact show that the protocol due to Reid et al. is *not* mafia fraud resistant. Another attack, which can be run in the case of prover-aborts, is a strong mafia attack, where the adversary simply takes over from the prover and authenticates. Both these attacks, but particularly the former one, enforce a need to upgrade the standard security of distance-bounding protocols from previous notions of mafia fraud resistance (without key-learning mafia fraud) to our stronger mafia fraud resistance notion and to strong mafia fraud resistance.
In Figure 40, we try to capture which properties we view as essential for even a minimally secure distance-bounding protocols (these are the standard security notions achieved by many of the distance-bounding schemes in the literature) and which properties are only necessary in higher-security scenarios. We refer to the future work section in section 7.3 for the discussion of several practical related issues. By comparing this figure with the results of our protocol assessment, practical protocols can be chosen such that they suit specific application scenarios.

Tools and Constructions. Though it remains an open question whether SimTF security can be achieved in a more efficient manner than we have shown in our construction, this seems doubtful, since the need for a back-door for the simulator seems inherent to the SimTF model. Furthermore, our construction also achieves the stronger guarantee of strSimTF security, which makes it suitable to high security settings.
The most important computational expense in our location private protocol is the exponential delay of the prover. Whereas it is possible to improve on the efficiency of the rest of the protocol, this exponential delay is necessary, in view of our impossibility results in Chapter 6.
Of the tools we developed, a very significant one is the key update compiler, which can be used either in its original form, to attain narrow-destructive privacy, or as a tool to achieve KLMF-security for schemes which are not resistant to key-learning attacks. We note that the strMF compiler —and its optimisations— can be easily deployed for most distance-bounding protocols in the literature, thus attaining full mafia fraud resistance and strong mafia fraud resistance. Of course, new constructions could also be designed from scratch to attain this property; however, by using our compiler, we can simply use an existing construction from the literature. We also note that the privacy model of Vaudenay [73] is the most widely-accepted privacy model for RFID privacy. Considering that an important application scenario of distance-bounding protocols is indeed RFID distance-bounding authentication, we note that a compiled distance-bounding protocol attained by means of our compiler would meet a widely recognised and standard security definition.
The tricks we outlined under Tools are also handy. Most of these are particularly useful to protocol designers or theoreticians who wish to extend our models (this is the case for the mafia and GameTF proof strategies, as well as for bypassing mafia

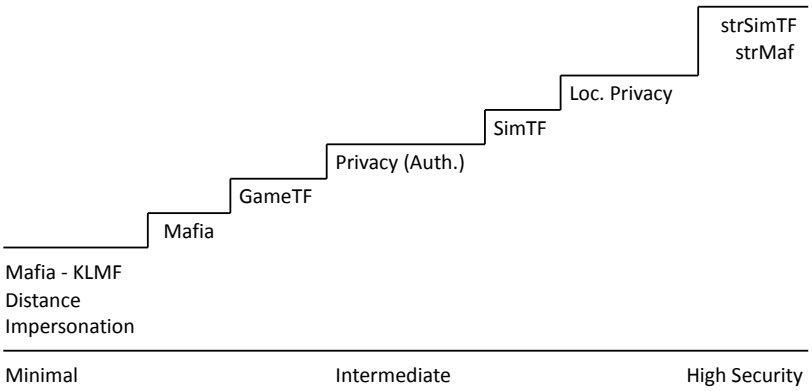

Figure 40: Practical relevance of security properties

fraud resistance). However, our trick of using mutual authentication as a trigger to update the prover's state is constructive and can be successfully used in protocol design to enable some control of the prover's behaviour by the verifier (since the verifier is often the computationally stronger party, with more resources, this seems a good way to defer some of the computation to it).

7.3 Conclusion and Future Work

This thesis represents a building block towards attaining a complete overview of provable security in distance-bounding protocols. In this section we conclude by showing how far this thesis comes to meeting the ultimate goal of completely describing distance-bounding security, and we outline directions for future work.

Models. On the one hand, we contribute towards this overview by setting up an exact and formal security model, disproving previous beliefs regarding the interdependency of the key properties in distance-bounding protocols: mafia fraud, terrorist fraud, distance fraud, and impersonation security. We also investigate aspects such as authentication privacy and location privacy, and further look into flavours of mafia and terrorist fraud attacks. By relating these notions we provide a cohesive and comprehensive picture of the security of distance-bounding protocols.
However, though our model is very comprehensive, we do not claim it is complete. Indeed, an important next step is to investigate a scenario where multiple provers interact with the verifier, and subsequently, the multiple-prover-multiple-verifier scenario. We note that the attack by Cremers et al. [22] shows that it is possible for one dishonest prover to commit distance fraud if it is in the presence of an honest prover within the verifier's proximity. Thus, an analysis of the multiple-prover-multiple-verifier case is paramount to understanding the effect of large-scale deployment of provers and verifiers. Furthermore, we note that our model leaves a few open questions such as: we know that the key update compiler does not preserve SimTF security. Does it, however, preserve GameTF security, which seems to be the notion that most protocols in the literature attain? Similarly, an interesting open question is whether we can design a location privacy achieving compiler that also preserves other security properties.
A further, very interesting direction concerns our timed/timeless channel model in Chapter 6: in particular, it would be interesting to investigate in how far our round-based results transfer to this different setting. Finally, a third direction would be to investigate distance-bounding in an asymmetric setting, where the prover and verifier — rather than sharing

a symmetric key K — possess a private/public key pair. This scenario is applicable to newer generations of RFID tags, which can run elliptic curve (EC) computations and can support public key cryptography.

Constructions. Our second main contribution in this thesis is providing concrete tools and constructions, which can be employed in practice to achieve properties such as: SimTF security, location privacy, KLMF and strMF security, and narrow-destructive privacy. As we have stressed before, key-learning attacks are very easily implementable in authentication, thus also in distance-bounding scenarios, and we believe that it is paramount to achieve KLMF resistance (see Fig. 40). Our KLMF compiler is particularly suitable towards that goal, since it enables us to reuse existing protocols in order to achieve better security properties. This also holds for our two other compilers, the strMF compiler and the key update compiler. We also provide proof techniques and constructive tricks that enable cryptographers to easily use our model and relate security notions, and that also enable designers to more easily build secure constructions.
An interesting and thus-far unexplored field for future work is to consider designing distance-bounding protocols that use the efficiency and highly-desirable properties of newer RFID hardware, which are able to perform EC computations at a lower cost than implementing HMACs. In view of recent discussions regarding the practical security of HMACs, we are further motivated to seek efficient authentication, and then efficient distance-bounding on dedicated EC processors.
There are a few other research directions one can consider. In particular, our proofs for the key update and resp. strMF compilers are not tight, since the adversary loses a polynomial factor in guessing which authentication key was used in a successful authentication attempt. We are on the one hand interested in tight proofs (if these exist), and on the other hand we should attempt to find also different compilers, which, perhaps at the cost of some computation efficiency, can be tightly reduced to the security properties of the underlying distance-bounding protocol. Another research direction is to design from scratch constructions attaining full mafia fraud and strMF security, and directly reducing their security to that of the underlying primitives. In this fashion we could obtain a much better bound than by employing the compiler (whose reduction is loose).

Implementations. Finally, most of our work has been theoretic. We note that the efficiency of our constructions is easily compared to that of other protocols in the literature; however, in order to be deployed in practice, distance-bounding protocols must be implemented on real-life hardware, such that the properties of the schemes are assessed. Already some ground work has been laid in this field by e.g. [42, 41, 70, 21] in the context of RFID, and the conclusion seems to be that distance-bounding can only be implemented in low-latency channels. An open question is whether different RFID architectures can support distance-bounding protocols and, in fact, *which* general architectures, RFID or otherwise, would support such schemes.

References

[1] Abyneh, M.R.S.: Security analysis of two distance-bounding protocols. In: RFID. Security and Privacy. Lecture Notes in Computer Science, vol. 7055, pp. 94–107. Springer-Verlag (2012)

[2] Armknecht, F., Sadeghi, A.R., Scafuro, A., visconti, I., Wachsmann, C.: Impossibility Results for RFID Privacy Notions. In: Gavrilova, M., Tan, C., Moreno, E. (eds.) Transactions on Computational Science XI, Lecture Notes in Computer Science, vol. 6480, pp. 39–63. Springer-Verlag (2010)

[3] Aumasson, J.P., Mitrokotsa, A., Peris-Lopez, P.: A Note on a Privacy-preserving Distance Bounding Protocol. In: 13^{th} International Conference on Information and Communications Security. Springer (2011)

[4] Avoine, G., Bingol, M.A., Karda, S., Lauradoux, C., Martin, B.: A formal framework for analyzing RFID distance bounding protocols. In: Journal of Computer Security - Special Issue on RFID System Security, 2010 (2010)

[5] Avoine, G., Lauradoux, C., Martin, B.: How secret-sharing can defeat terrorist fraud. In: Proceedings of the Fourth ACM Conference on Wireless Network Security WISEC 2011. pp. 145–156. ACM Press (2011)

[6] Avoine, G., Tchamkerten, A.: An efficient distance bounding RFID authentication protocol: Balancing false-acceptance rate and memory requirement. In: 12^{th} International Conference on Information Security (ISC) 2009. Lecture Notes in Computer Science, vol. 5735, pp. 250–261. Springer-Verlag (2009)

[7] Bellare, M., Goldreich, O.: Proving computational ability. `http://www.wisdom.weizmann.ac.il/~oded/PS/poa.ps` (1992)

[8] Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Advances in Cryptology — EUROCRYPT 2000. Lecture Notes in Computer Science, vol. 1807, pp. 139–155. Springer-Verlag (2000)

[9] Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Advances in Cryptology — CRYPTO '93. Lecture Notes in Computer Science, vol. 773, pp. 232–249. Springer-Verlag (1994)

[10] Bogdanov, A., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J., Seurin, Y.: Hash functions and RFID tags: Mind the gap. In: Cryptographic Hardware and Embedded Systems - CHES 2008. Lecture Notes in Computer Science, vol. 5154, pp. 283–299. Springer-Verlag (2008)

[11] Boureanu, I., Mitrokotsa, A., Vaudenay, S.: On the pseudorandom function assumption in (secure) distance-bounding protocols (2012)

[12] Brands, S.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press (2000)

[13] Brands, S., Chaum, D.: Distance-bounding protocols. In: Advances in Cryptology — EUROCRYPT'93. pp. 344–359. Lecture Notes in Computer Science, Springer-Verlag (1993)

[14] Bringer, J., Chabanne, H.: Trusted-HB: A low-cost version of HB $^{+}$ secure against man-in-the-middle attacks. Transactions on Information Theory 54(9), 4339–4342 (2008)

[15] Bussard, L., Bagga, W.: Distance-bounding proof of knowledge to avoid real-time attacks. IFIP International Federation for Information Processing, vol. 181, pp. 222–238. Springer-Verlag (2005)

[16] Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Advances in Cryptology — EUROCRYPT 2001. Lecture Notes in Computer Science, vol. 2045, pp. 93–118. Springer-Verlag (2001)

[17] Carluccio, D., Kasper, T., Paar, C.: Implementation details of a multi purpose ISO 14443 rfidtool. In: Printed handout of Workshop on RFID Security - RFIDSec 06 (July 2006)

[18] Chandran, N., Goyal, V., Moriarty, R., Ostrovsky, R.: Position based cryptography. In: Advances in Cryptology — CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 391–407. Springer-Verlag (2009)

[19] Chatmon, C., van Le, T., Burmester, M.: Secure anonymous RFID authentication protocols. Florida State University, Department of Computer Science, Tech Report (2006)

[20] Clulow, J., Hancke, G.P., Kuhn, M.G., Moore, T.: So near and yet so far: Distance-bounding attacks in wireless networks. In: European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks. Lecture Notes in Computer Science, vol. 4357, pp. 83–97. Springer-Verlag (2006)

[21] Cole, P.H., C.Ranasinghe, D.: Networked RFID Systems and Lightweight Cryptography. Springer-Verlag (2008)

[22] Cremers, C., Rasmussen, K.B., Čapkun, S.: Distance hijacking attacks on distance bounding protocols. Cryptology ePrint Archive, Report 2011/129 (2011), http://eprint.iacr.org/2011/129.pdf

[23] Deng, R.H., Li, Y., Yung, M., Zhao, Y.: A new framework for RFID privacy. In: Proceedings of the 15^{th} European Symposium on Research in Computer Security, (ESORICS'10). Lecture Notes in Computer Science, vol. 6514, pp. 1–18. Springer-Verlag (2010)

[24] Desmedt, Y.: Major security problems with the 'unforgeable' (feige)-fiat-shamir proofs of identity and how to overcome them. In: SecuriCom. pp. 15–17. SEDEP Paris, France (1988)

[25] Drimer, S., Murdoch, S.J.: Keep your enemies close: distance bounding against smartcard relay attacks. In: Proc. of the 16-th USENIX Security Symposium on USENIX Security Symposium, article no. 7. ACM (2007)

[26] Duc, D., Kim, K.: Securing HB+ against GRS man-in-the-middle attack. In: Symposium on Cryptography and Information Security (SCIS). The Institute of Electronics, Information and Communication Engineers (2007)

[27] Dürholz, U., Fischlin, M., Kasper, M., Onete, C.: A formal approach to distance bounding RFID protocols. In: Proceedings of the 14^{th} Information Security Conference ISC 2011. Lecture Notes in Computer Science, vol. 7001, pp. 47–62. Springer-Verlag (2011)

[28] Dwork, C., Lotspiech, J.B., Naor, M.: Digital signets: Self-enforcing protection of digital information (preliminary version). In: STOC. pp. 489–498 (1996)

[29] Feldhofer, M., Rechberger, C.: A case against currently used hash functions in RFID protocols. In: On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. LNCS, vol. 4277, pp. 372–381. Springer-Verlag (2006)

[30] Fischlin, M., Onete, C.: RFID distance-bounding: What is wrong and how to fix it. In: accepted at 5^{th} MPICC Interdisciplinary Conference on Current Issues in IT Security. Interdisciplinary series of the Max Planck Institute for Foreign and International Criminal Law

[31] Fischlin, M., Onete, C.: Distance-bounding and terrorist fraud: Simulation and game-based notions. In Submission (2012)

[32] Fischlin, M., Onete, C.: Provably secure distance-bounding: an analysis of prominent protocols. Cryptology ePrint Archive, Report 2012/128 (2012), http://eprint.iacr.org/2012/128.pdf

[33] Francillon, A., Danev, B., Čapkun, S.: Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars (2010), http://eprint.iacr.org/2010/332

[34] Francis, L., Hancke, G., Mayes, K., Markantonakis, K.: Practical NFC peer-to-peer relay attack using mobile phones. In: RFIDSec'10 Proceedings of the 6^{th} international conference on Radio frequency Identification: security and privacy issues. pp. 47–62. Springer-Verlag (2010)

[35] Gilbert, H., Robshaw, M., Sibert, H.: An active attack against HB+ - a provably secure lightweight authentication protocol. Cryptology ePrint Archive, Report 2005/237 (2005), http://eprint.iacr.org/2005/237.pdf

[36] Goldreich, O., Pfitzmann, B., Rivest, R.L.: Self-delegation with controlled propagation - or - what if you lose your laptop. In: Advances in Cryptology — CRYPTO'98. Lecture Notes in Computer Science, vol. 1462, pp. 153–168. Springer-Verlag (1998)

[37] H-Security, T.: Chip-based ID cards pose security risk at airports. `http://www.h-online.com/security/news/item/Chip-based-ID-c`
`at-airports-905662.html` (2010)

[38] Haataja, K., Toivanen, P.: Two practical man-in-the-middle attacks on bluetooth secure simple pairing and countermeasures. Transactions on Wireless Communications 9(1), 384–392 (2010)

[39] Hancke, G.P.: A practical relay attack on ISO 14443 proximity cards. http://www.cl.cam.ac.uk/gh275/relay.pdf (2005)

[40] Hancke, G.P.: Practical Attacks on Proximity Identification Schemes (2006)

[41] Hancke, G.P.: Design of a secure distance-bounding channel for RFID. Journal of Network and Computer Applications (2010)

[42] Hancke, G.P., Kuhn, M.G.: An RFID distance bounding protocol. In: Conference on Security and Privacy for Emergency Areas in Communication Networks (SecureComm) 2005. pp. 67–73. IEEE (2005)

[43] Hermans, J., Pashalidis, A., Vercauteren, F., Preneel, B.: A new RFID privacy model. In: Proceedings of the 16^{th} European Symposium on Research in Computer Security, (ESORICS) 2011. Lecture Notes in Computer Science, vol. 6879, pp. 568–587. Springer-Verlag (2011)

[44] Hlaváč, M., Tomáč, R.: A Note on the Relay Attacks on e-Passports (2007), http://eprint.iacr.org/2007/244.pdf

[45] Hopper, N.J., Blum, M.: Secure human identification protocols. In: Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security (ADVCRYPTO) 2001. Lecture Notes in Computer Science, vol. 2248, pp. 52–66. Springer-Verlag (2001)

[46] Juels, A.: RFID security and privacy: a research survey. IEEE Journal on Selected Areas in Communications 24(2), 381–394 (2006)

[47] Juels, A., Weis, S.A.: Defining strong privacy for RFID. In: International Conference on Pervasive Computing and Communications - Workshops (PerCom Workshops). pp. 342–347. IEEE (2007)

[48] Kfir, Z., Wool, A.: Picking virtual pockets using relay attacks on contactless smartcard systems. In: Conference on Security and Privacy for Emergency Areas in Communication Networks – SecureComm 2005. pp. 47 – 58. IEEE (2005)

[49] Kim, C.H., Avoine, G.: RFID distance bounding protocol with mixed challenges to prevent relay attacks. In: Proceedings of the 8th International Conference on Cryptology and Networks Security (CANS) 2009. Lecture Notes in Computer Science, vol. 5888, pp. 119–131. Springer-Verlag (2009)

[50] Kim, C.H., Avoine, G., Koeune, F., Standaert, F., Pereira, O.: The swiss-knife RFID distance bounding protocol. In: Information Security and Cryptology (ICISC) 2008. pp. 98–115. Lecture Notes in Computer Science, Springer-Verlag (2008)

[51] Koblitz, N., Menezes, A.: Another look at HMAC. Cryptology ePrint Archive, Report 2012/074 (2012), http://eprint.iacr.org/2012/074

[52] Lee, Y., Sakiyama, K., Batina, L., Verbauwhede, I.: Elliptic curve based security processors for RFID. In: IEEE Transactions on Computers. vol. 57, pp. 1514–1527. IEEE (2008)

[53] Leinweber, L., Papachristou, C., Wolff, F.G.: A case against currently used hash functions in rfid protocols. In: 2009 IEEE International Conference on Computer Design. pp. 372–377. IEEE (2009)

[54] Leng, X., Mayes, K., Markantonakis, K.: HB-MP+ protocol: An improvement on the HB-MP protocol. In: International Conference on RFID. pp. 118–124. IEEE Computer Society Press (2008)

[55] Levi, A., Çetintas, E., Aydos, M., Koç, Çetin Kaya., Çaglayan, M.U.: Relay attacks on bluetooth authentication and solutions. In: International Symposium Computer and Information Sciences (ISCIS) 2004. Lecture Notes in Computer Science, vol. 3280, pp. 278 – 288. Springer-Verlag (2004)

[56] Ma, C., Li, Y., Deng, R.H., Li, T.: RFID privacy: relation between two notions, minimal condition, and efficient construction. In: Proceedings of the 16^{th} ACM conference on Computer and communications security, (CCS'09). pp. 54–65. ACM (2009)

[57] Mitrokotsa, K., Onete, C., Vaudenay, S.: Location leakage in distance bounding: Why location privacy doesn't work. In Submission (2012)

[58] Mitrokotsa, K., Onete, C., Vaudenay, S.: Mafia fraud attack against the RČ distance-bounding protocol. In Submission (2012)

[59] Ng, C.Y., Susilo, W., Mu, Y., Safavi-Naini, R.: RFID privacy models revisited. In: Proceedings of the 13^{th} European Symposium on Research in Computer Security, (ESORICS) 2008. Lecture Notes in Computer Science, vol. 5283, pp. 251–266. Springer-Verlag (2008)

[60] Onete, C.: Key updates for RFID distance-bounding protocols: Achieving narrow-destructive privacy. Cryptology ePrint Archive, Report 2012/165 (2012), http://eprint.iacr.org/2012/165

[61] Onete, C.: Mafia fraud resistance revisited: Key-learning attacks. In Submission (2012)

[62] Oren, Y., Wool, A.: Relay attacks on RFID-based electronic voting systems. Cryptology ePrint Archive, Report 2009/442 (2009), http://eprint.iacr.org/2009/422.pdf

[63] Ouafi, K., Overbeck, R., Vaudenay, S.: On the security of HB# against a man-in-the-middle attack. In: Advances in Cryptology — Asiacrypt 2008. Lecture Notes in Computer Science, vol. 5350, pp. 108–124. Springer (2008)

[64] Paise, R.I., Vaudenay, S.: Mutual authentication in RFID: Security and privacy. In: Proceedings on the 3^{rd} ACM symposium on information, computer and communications security (ASIACCS) 2008. pp. 292–299. ACM (2008)

[65] Pelechrinis, K., Koufogiannakis, C., Krishnamurthy, S.V.: On the Efficacy of Frequency Hopping in Coping with Jamming Attacks in 802.11 Networks. IEEE Transactions on Wireless Communications 9(10), 3258–3271 (October 2010)

[66] Ranganathan, A., Tippenhauer, N.O., Singelée, D., koric, B., Capkun, S.: Design and Implementation of a Terrorist Fraud Resilient Distance Bounding System. In: Foresti, S., Martinelli, F., Yung, M. (eds.) 2012nd European Symposium on Research in Computer Security (ESORICS 2012). Lecture Notes in Computer Science, vol. 7459, pp. 415–432. Springer-Verlag, Pisa,Italy (2012)

[67] Rasmussen, K.B., Čapkun, S.: Realization of RF distance bounding. USENIX Security Symposium (2010)

[68] Rasmussen, K.B., Čapkun, S.: Realization of RF Distance Bounding. In: USENIX 2010. pp. 389–402 (2010)

[69] Rasmussen, K., Čapkun, S.: Location privacy of distance bounding. In: Proceedings of the Annual Conference on Computer and Communications Security (CCS). pp. 149–160. ACM (2008)

[70] Reid, J., Nieto, J.M.G., Tang, T., Senadji, B.: Detecting relay attacks with timing-based protocols. In: Proceedings of the 2^{nd} ACM symposium on information, computer and communications security (ASIACCS) 2007. pp. 204–213. ACM Press (2007)

[71] Sadeghi, A.R., Visconti, I., Wachsmann, C.: Anonymizer-enabled security and privacy for RFID. In: Advances in Cryptology — Asiacrypt 2008. Lecture Notes in Computer Science, vol. 5888, pp. 292–299. Springer-Verlag (2009)

[72] Spil, D., Bittau, A.: Bluesniff: Eve Meets Alice and Bluetooth. In: Proceedings of the 1st USENIX Workshop on Offensive Technologies (WOOT) 2007. USENIX Association Berkley, CA, USA (2007)

[73] Vaudenay, S.: On privacy models for RFID. In: Advances in Cryptology — Asiacrypt 2007. Lecture Notes in Computer Science, vol. 4883, pp. 68–87. Springer-Verlag (2007)

[74] Čapkun, S., Defrawi, K.E., Tsudik, G.: Group distance bounding protocols. In: Proceedings of the 4^{th} international conference on Trust and trustworthy computing (TRUST) 2011. Lecture Notes in Computer Science, vol. 6740, pp. 302–312. Springer-Verlag (2011)

[75] Wenger, E., Hutter, M.: A hardware processor supporting elliptic curve cryptography for less than 9 kges. In: Proceedings of the 10^{th} Smart Card Research and Advanced Applications (CARDIS) 2011. Lecture Notes in Computer Science, vol. 7079, pp. 182–198. Springer-Verlag (2011)

[76] Yung, M.: Zero-knowledge proofs of computational power. In: Advances in Cryptology — EUROCRYPT '89. Lecture Notes in Computer Science, vol. 434, pp. 196–207. Springer-Verlag (1990)

[77] Zhang, E.Y., Kitsos, P.: Security in RFID and Sensor Networks. CRC Press, Taylor & Francis Group (2009)